MASTER EN SCIENCES INFORMATIQUES

Analyse et visualisation des réseaux sociaux : cas de Twitter

AKITIO TIOFACK, Giudice

Award date:
2011

Awarding institution:
Universite de Namur

Facultés Universitaires Notre Dame De la Paix (FUNDP Namur)
2e Master - Faculté d'Informatique - Année académique 2010 - 2011

Mémoire présenté en vue de l'obtention du diplôme de Master en Sciences Informatiques.

Analyse et visualisation des réseaux sociaux : cas de Twitter

Giudice AKITIO TIOFACK
Promoteur : Mme Monique NOIRHOMME

31 août 2011

Résumé

Actuellement, les réseaux sociaux occupent une place important dans notre société. La plupart de ces réseaux sociaux sont dédiés à des usages privés. Les adminnistrations commencent à s'y intérésser, c'est le cas par exemple de l'administration française qui aimerait se munir d'un réseaux social et disposer d'un outil lui permettant d'avoir une idée de sa réputation, de son image sur internet.
Dans ce mémoire, nous présentons un projet d'analyse de réseaux sociaux pour l'administration. De ce projet, découle la mise sur pieds d'un prototype d'outil de gestion de réputation sur le réseau social Twitter : *Evarist*. Lors de la conception d'*Evarist*, les méthodes de l'Analyse Formelle de Concepts ont été utilisées pour la classification de l'information. Etant donné que nous avons travaillé dans l'environnement R, un package R a donc été développé, implémentant les fonctions nécessaires de construction de Treillis de concepts. Ce package nommé *Galois* sera présenté de manière détaillé. L'objectif final sera de proposer l'intégration de ce package à la communauté des développeurs R.

Mots clés

Réseau social, Analyse Formelle de Concepts, Treillis de concepts, *R*.

Abstract

Today, social networks play an important role in our society. Most of these social networks are dedicated to private use. Now public administration begin to be interest. It is the case for French administration which want to carry a social network. It also want to have an tool which would help they to get an idea of its reputation, its image on the internet.
In this paper, we present one social network analysis project for administration. This project allowed us to implement a prototype tool for reputation management on the social network Twitter. This tool is called *Evarist*. In the design of Evarist, we used methods of formal concept analysis for classification of data. Since we worked in the R environment, an R package has been developped. It implements the functions we need to build concept lattice. This package is named *Galois* and will be presented in detail. The ultimate gaol of our work will be to suggest the integration of this package to the developper community R.

Keywords

Social network, Formal Concept Analysis, Concept Lattice, *R*.

Avant-propos

Je tiens à adresser mes sincères remerciements toutes les personnes qui ont contribué directement ou indirectement à la réalisation de ce mémoire :

- En premier lieu, Mme Noirhomme, professeure aux FUNDP. En tant que Directrice du mémoire, elle m'a guidé et a toujours été disponible.
- Je remercie particulièrement Mme Marie-Aude Aufaure, Etienne Cuvelier mon encadreur de stage et tout le personnel du Laboratoire MAS (Mathématiques Appliquées aux Systèmes) de l'Ecole Centrale Paris pour leur encadrement.
- Les professeurs des FUNDP pour tous les enseignements qu'ils m'ont transmis durant mes années d'étude.
- Le service sociale des étudiants des FUNDP pour leur soutien tout au long de ma formation.
- A tous mes amis qui m'ont soutenu durant ces années difficiles.
- A mon tonton Louis Stephen Tedajio pour ses conseils.
- Enfin à toute ma famille et particulièrement à mes parents pour leur confiance et leur aide tout au long de mes études.

Je dédie ce mémoire à mes feus grand-mères Marthe Tsapé et Jeanne Dongmo.

Table des matières

Introduction

Présent un peu partout, internet occupe une place de choix dans toutes nos activités. A travers diveres applications disponibles, les internautes ont tendance à construire et à faire vivre entre eux des relations, des échanges. Ceci se caractérise principalement par la création en ligne de groupes de personnes partageant un certain nombre de centre d'intérêts. Les réseaux sociaux deviennent applicables sur la toile. On assiste ainsi à un véritable essor des réseaux sociaux sur internet. Souvent utilisés pour faire des rencontres, retrouver des connaissances ou partager des expériences, ces réseaux sociaux connaissent un engouement grandissant. C'est un domaine qui a connu une évolution très rapide. Loin d'être réservé à un usage privé, les réseaux sociaux en ligne sont aussi utilisés dans des buts professionels et par des entreprises. Alors que le nombre de réseaux sociaux en ligne est sans cesse en augmentation, il devient important pour les internautes, aussi bien les particuliers que les entreprises de pouvoir gérer leur image, leur réputation sur la toile. Il existe un grand nombre d'outils de gestion de la réputation en ligne. Ce mémoire s'articule autour d'un nouvel outil de gestion de l'e-réputation. Cet outil dérive d'un projet d'analyse de réseaux sociaux pour l'administration : le projet ARSA [1].

Ce mémoire s'inscrit dans le cadre de la réalisation du projet ARSA. Ce projet se veut être le fruit d'une collaboration entre l'Ecole Centrale Paris, SAP et le data center Euclide. Il est né de l'appel d'offre de l'administration française et a pour but la mise sur pieds d'un réseau social pour l'administration. C'est l'Ecole Centrale Paris, lieu où nous avons effectué notre stage qui se charge du développement des modules d'analyse.

Ce mémoire se répartit en plusieurs chapitres :

1. **Les réseaux sociaux** :Le premier chapitre est consacré à la définition du concept de réseau social. Il présente de manière brève l'historique des réseaux sociaux sur le web. De plus, nous y caractérisons les différents

1. Analyse des Réseaux Sociaux de l'Administration

types de réseaux sociaux en ligne que nous pouvons rencontrés. Notre travail étant basé sur Twitter, une section de ce chapitre est consacré à sa présentation.

2. **L'analyse des réseaux sociaux : formalisation et approche par la théorie des graphes** : après avoir défini la notion de réseau social, nous présentons dans ce chapitre les outils mathématiques mécessaires à sa représentation et à son analyse.
3. **L'Analyse Formelle de concepts (AFC)** : ce chapitre nous permet de voir comment l'AFC utilisé en général comme méthode d'extraction de connaissances peut être, de part le fait que c'est une approche de découverte et de structuration hiérarchique de connaissance, un outil intéressant d'analyse de réseaux sociaux. Ce chapitre présente principalement la notion de treillis et différents algorithmes de construction de ces derniers.
4. **Projet ARSA** : ce troisième chapitre rappel l'origine et le but du projet. On y présente les différents acteurs qui participent à la réalisation de ce projet.
5. **Evarist** : ce chapitre présente la notion d'e-réputation. Evarist est un prototype d'outil d'e-réputation implémenté dans le cadre du projet ARSA. Nous expliquons en détail cet outil qui a été développé dans l'environnement R. Evarist est une première réponse aux attentes définies par le projet ARSA.
6. **L'environnement R** : ce projet a été développé en R. Ce chapitre a pour objet la présentation de cet environnement. Nous y présentons ses caractéristiques, ses points forts et ses points faibles.
7. **Le package Galois** : L'un des objectifs de notre travail était la réalisation d'un package R comportant les différentes fonctions de construction de Treillis de Galois. Ce package est le principal module utilisé pour le développement du projet Evarist. Ce chapitre présente dans un premier temps la méthodologie de construction d'un package R. Dans un second temps, il détaille les différentes fonctions que nous avons implémenté dans le cadre du projet Evarist.
8. **Application** : dans ce dernier chapitre, sur base d'un exemple, nous illustrons les résultats obtenus lors de l'exécution des modules développés dans le cadre du projet Galois.

Chapitre 1

Les réseaux sociaux

1.1 Origine de la notion de réseau

Le mot "réseau" [Nic06] tire ses racines du mot "réseuil" qui s'écrit encore "résel", lui même tiré du mot latin *retiolus* décliné en *rete/retis*, qui signifie *filet* et désigne un antrelacs de lignes, représentant ainsi un tissu utilisé par les chasseurs, mais aussi par les femmes pour retenir leurs cheveux [Nic06]. Tandis que le mot latin *retis* permet de dériver le mot français "rets" et l'adjectif "réticulaire", en français, le mot réseul, se transforme dès la fin du XVIIème siècle en "rézeau", puis plus tard on dérivera le mot "réseau". La première édition du *Dictionnaire de l'académie française (1694)* définira un réseau comme étant un "ouvrage de fil ou de soye, fait par petites mailles en forme de rets". Ce qui désigne aussi bien les pièges des chasseurs que les coiffes des dames. Mais au XVIIème siècle, le réseau, définira l'entrecroisement des fibres.
Par la suite, nous avons assisté à l'extension de la notion de réseau du registre de textile au registre médical. Dans un premier temps, il désignait l'appareil sanguin, ensuite il était utilisé pour désigner aussi le système nerveux. Avec cette extension qui s'ajoute à son sens premier c'est-à-dire l'entrelacement, s'ajoute implicitement celui de la circulation dont le réseau est le support.
C'est au XIXème siècle que ce dernier sens devient explicite, ceci avec l'usage de plus en plus répandu de la notion de réseau pour désigner l'ensemble des chemins, des routes, puis des voies ferrées. Au milieu du XIXème siècle on a commencé à avoir recours à la notion de réseau pour désigner des ensembles d'individus ainsi que les relations qu'ils entretiennent entre eux. C'est à ce moment que les sciences sociales s'en emparent pour désigner ce qui était connu jusque là comme des groupes, des cercles. C'est dans un article de l'anthropologue britanique John A. Barnes [Bar54] que la notion de "réseau

social" fait sa première apparition [Mer03] .

1.2 Concept du réseau social

L'organisation sociale structure la vie en congrégation. Les hommes ont toujours tendance à se regrouper, à vivre en communautés plus ou moins structurées. Ainsi, une famille, des amis, une équipe de travail peuvent constituer des réseaux sociaux. Dès lors, un réseau social peut se définir comme étant un ensemble d'entités telles que des individus, des entreprises, des organisations reliées entre elles par des liens créés à travers des interactions sociales. Dans le même ordre d'idée, Wasserman et Faust [WF94] décrivent un réseau social comme étant un ensemble de relations entre entités sociales. Ces relations peuvent par exemple être des relations d'amitié, de collaboration. Certains auteurs définissent le réseau social comme étant une communauté regroupant des utilisateurs sur un thème.
Aujourd'hui, le terme réseau social s'applique en particulier au domaine de l'Internet. Car l'homme va chercher à y transposer le calque de sa vie, d'y trouver une forme d'organisation, d'association autour de mêmes idéaux, afin d'élargir son réseau social naturel [1] qu'il s'est construit dans la vie de tous les jours. Dès lors, le réseau social "désigne un site web qui, dans un domaine quelconque, fédère des individus et facilite leurs échanges d'information, d'images, de photos ..." [2] . Nous remarquons que l'impact des réseaux sociaux sur les usages des systèmes d'information, qu'ils soient dans le domaine du privé ou du professionnel est très significatif [3]. On note dès lors une évolution exponentielle des réseaux sociaux numériques. Les réseaux sociaux sur Internet sont devenus des outils incontournables pour les entreprises et bientôt pour les administrations publiques [4]. A l'heure actuelle, toute entreprise qui souhaite assurer le développement de son économie numérique, quelle que soit sa taille et son secteur d'activité, se doit de connaître et de s'appuyer sur les réseaux sociaux. Pour créer un réseau, il est primordial de pouvoir mettre en relation plusieurs entités. Cette activité est connue sous le nom de "réseautage social". Le résautage social se rapporte à l'ensemble des moyens mis en oeuvre pour relier des personnes physiques, morales entre

1. Nous considérons qu'un réseau social est dit naturel lorsqu'il n'est pas crée sur base d'outils technologiques, d'internet. C'est le cas par exemple d'un réseau constitué sur base de lien familliaux.

2. www.web-libre.org/dossiers/reseaux-sociaux,4931.html, dernière visite le 16 février 2011

3. La question qui en découle est celle de savoir si pour une expression du moi, le réseau naturel n'est-il pas suffisant.

4. Exemple avec le projet ARSA en France

elles. Avec l'apparition d'internet, cette notion recouvre aussi les applications informatiques liées à Internet qui servent à constituer un réseau social [5] .

Le Web2.0 [6] est le principal catalyseur de l'essor des réseaux sociaux sur Internet. C'est un concept d'utilisation d'internet qui a pour but de valoriser l'utilisateur et ses relations avec les autres. En effet, il dispose de plusieurs caractéristiques [Wes07] qui favorisent la mise en oeuvre des réseaux sur Internet. Une caractéristique est l'utilisation de l'intelligence collective. Les utilisateurs ont la possibilité d'ajouter du contenu au site, ce qui peut conduire à une amélioration de la qualité du site. Les internautes contribuent ainsi à la gestion du site. C'est donc toute une communauté qui participe à sa maintenance. Le web est alors basé sur les données, l'information est plus importante que le site lui-même. Plus les données sont facilement accessibles, plus elles ont de chances d'être utilisées et plus elles sont visibles.
Avant de pousser un peu plus loin l'analyse de la structure des réseaux sociaux, il nous semble intéressant de présenter une brève historique de l'apparition de ces réseaux sociaux sur le web.

1.3 Historique des réseaux sociaux sur le web

On dénombre à nos jours une multitude de réseaux sociaux. Le réseau ARPANET peut être vu comme le prémisse nécessaire à l'apparition des réseaux sociaux en ligne. En effet, ce dernier a vu le jour en 1969, il avait pour but d'interconnecter différents réseaux. Ce réseau créé par les service militaires américains utilisait la communication par paquet [7].
La figure 1.1 [8] est une image représentative et non exhaustive de l'apparition des réseaux sociaux .

Comme le présente la figure 1.1, on note que c'est en 1988 que AOL a procédé au lancement d'une plateforme 'sociale' aux profils associés par intérêts. En mai 2003, LinkedIn est lancé, et quelques mois plus tard, en juillet MySpace voit le jour. En octobre de la même année, MySpace ajoute des commentaires sur les photos individuelles, et en décembre, il ajoute sa propre messagerie. L'an 2004 voit la naissance de Facebook et un an plus

5. www.old.saferinternet.org/ww/fr/pub/insafe/safety_ issues/faqs/social_ networking.htm, dernière viste le 19 février 2011

6. Terme suggéré par Dale Dougherty lors d'un brainstorming entre O'Reilly et Mediallive International [Or05]

7. www.dictionnaire.phpmyvisites.net/definition-ARPANET-4117.htm, dernière visite le 19 février 2011

8. www.histoire-cigref.org/blog/d-arpanet-aux-reseaux-sociaux-d-aujourdhui/, dernière visite le 19 février 2011

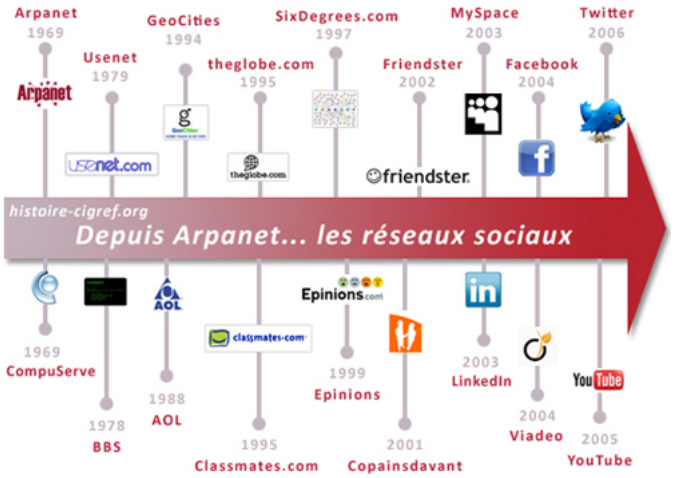


FIGURE 1.1 – Apparition des réseaux sociaux

tard, nait youtube. En 2006 par contre, on assiste au lancement de Twitter, qui initialement est un outil de réseau social et de microbloggage qui permet à l'utilisateur d'envoyer gratuitement des messages brefs, appelés tweets, au départ par sms et ensuite par internet ou par messagerie instantanée [9].

1.4 Fonctionnement des réseaux sociaux

Malgré le nombre sans cesse croissant de réseaux sociaux, force est de constater que tous sont basés sur un même principe de fonctionnement. En effet, le fonctionnement est très similaire d'un réseau à l'autre. Chaque membre crée un profil en indiquant un certain nombre de données personnelles et de centres d'intérêt. Puis, pour certains sites l'internaute doit inviter d'autres personnes à rejoindre son réseau, une liaison entre lui et les autres se créént si chaque partie donne son accord, les échanges d'informations vont dans les deux sens ; les liens sont symétriques. Pour d'autres sites l'utilisateur n'a pas besoin d'envoyer d'invitation pour constituer un réseau, les autres membres du site peuvent faire le choix de suivre les publications de ce dernier sans demander son autorisation. Les relations entre internautes dans ce cas sont asymétriques. Les sites comme *Google+* et *Twitter* proposent des relations asymétriques.
Dans le cas des sites utilisant des relations symétriques, que ce soit dans une

9. www.pascalbeauchesne.wordpress.com/2007/09/05/historique-des-reseaux-sociaux-de-aol-a-facebook/, dernière visite 19 février 2011

communauté d'amis, de parents, de voisins, ou de partenaires commerciaux, un premier ensemble d'acteurs (les fondateurs) envoie des messages invitant des membres à se réunir. Les nouveaux membres à leur tour envoient le même message à d'autres membres. Ce processus va se répéter, accroissant ainsi le nombre d'intervenants et de liens dans le réseau. Pour illustrer cela, nous pouvons citer comme exemple le site de rencontre *classmates.com* qui a pour but premier de faciliter la recherche de ses anciens camarades de classe, *Myspace* qui se focalise autour de la musique et des vidéos, ou encore *Facebook* pour ne citer que ceux-là.

1.5 Les types de réseaux sociaux

Les réseaux sociaux sont présents dans tous les secteurs d'activités. Ceci est dû au fait qu'un réseau social regroupe des entités partageant la même passion pour une activité quelconque. Il existe plusieurs critères de classification des réseaux sociaux. Les réseaux peuvent être ouverts ou sur invitation, implicites ou explicites, généralistes ou spécialisés, "grand public" ou d'entreprise.

1.5.1 Les réseaux sociaux implicites et explicites

Dans certains sites, il arrive que plusieurs utilisateurs collaborent pour la rédaction d'un article, d'un projet. C'est le cas par exemple de *Wikipedia*. Ces sites qui sont orientés en général autour des contenus font parties des réseaux sociaux dits *implicites*. Ces réseaux se constituent sur la base d'une activité réelle. Ces réseaux peuvent se déssiner en considérant les profils des différents utilisateurs. Les utilisateurs ayant des intérêts communs auront tendance à intervenir sur les mêmes sujets.
Par contre, les *réseaux sociaux explicites* sont des sites construits pour et par les utilisateurs [Tor06], ce sont les sites dans lesquels on nous propose explicitement des connections et où nous avons la possibilité de choisir formellement les éléments qui constitueront notre réseau. Dans cette catégorie on retrouve des réseaux tels que *Facebook* et *LinkedIn*. Ce type de réseau est utilisé par exemple lorsqu'on recherche des amis ou lorsqu'on veut se faire connaître.

1.5.2 Les réseaux sociaux généralistes et spécialisés

Les réseaux sociaux généralistes ont pour vocation première de permettre aux internautes de rester en contact [10]. Comme *réseaux sociaux généralistes*, nous citons *Facebook* et *MySpace*. Nous pouvons aussi y inclure *Twitter*, bien que la vocation première de ce dernier soit le micro-blogging [11]. Ils sont supposés ne pas avoir d'orientation prédéfinie [12]. Les communautés que nous rencontrons ici sont potentiellenemt grandes.

Contrairement aux réseaux généralistes, les *réseaux spécialisés* sur un intérêt commun entre les membres. Ils ont la particularité de proposer un thème spécifique à la base. Les réseaux sociaux pour étudiants [13] , les réseaux sociaux de famille [14] , les sites de rencontre, les résaux sociaux de proximité sont des réseaux sociaux spécialisés. Nous pouvons en citer quelques uns : les réseaux sociaux pour fans de musiques [15] , pour voisins [16] ,pour utilisateurs de train [17] , et même pour amateurs de chiens ou de chats.

1.5.3 Les réseaux sociaux "grands publics"

Par réseau "grand public", nous entendons les réseaux dédicacés au public.Ils sont principalement utilisés par les personnes des individus, qui vont se créer une page de profil afin d'y dévoiler des informations personnelles. Ces réseaux vont permettre aux individus de créer des communautés, de rencontrer des personnes avec les mêmes centres d'intérêt, de créer des groupes d'ami par exemple.

1.5.4 Les réseaux sociaux d'entreprise

Depuis le début de l'année 2010 on remarque une augmentation constante du nombre de réseaux sociaux d'entreprise. Les réseaux d'entreprise représentent des réseaux intra-organisationnels et inter-organisationnels. Calqué sur le modèle des réseaux sociaux publics, les réseaux sociaux d'entreprise ont pour

10. www.david.fayon.free.fr/stock/reseaux-sociaux.htm
11. www.commentquoi.com/reseaux-sociaux-partout-pour-tous-part-1-8675611.html, dernière visiste le 20 février 2011
12. www.marketing20.fr/marketing-communautaire-marketing-social/trop-de-reseaux-sociaux-fin-des-reseaux-sociaux/, dernière visiste le 20 février 2011
13. Exemple : www.reseaucampus.com
14. Exemple : www.famicity.com
15. Exemple : www.overbooke.com
16. Exemple : www.voisineo.com
17. Exemple : www.idtgvandco.com

objectif de permettre aux employés d'échanger avec leurs collègues par le biais de messages. Ces messages peuvent être privés ou publics. Le réseau social d'entreprise se veut avant tout être un outil collaboratif qui permet aux employés d'améliorer leur efficacité et de gérer leur image au sein de l'entreprise. Il est aussi perçu comme un outil de communication interne. Il minimise le rapport hiérarchique. Un employé peut, via cet outil présenter ses idées, faire connaître ses compétences en faisant des publications. Dans une entreprise, il est aussi considéré comme un outil de management, car les employés travaillant dans le même domaine de compétence peuvent partager leurs expériences quelque soit leur situation géographique [18] . L'intérêt de ces réseaux est d'établir des contacts d'affaires, de développer des partenariats, trouver des investiseurs. En général, ils sont de véritables atouts pour booster le business d'une entreprise [19] . C'est dans cette optique que Bertrand Duperrin dira que "S'agissant de bénéfices, un réseau social d'entreprise vient combler un vide. Aujourd'hui, les entreprises sont expertes dans la façon de dupliquer des process, mais elles sont moins performantes quand il s'agit de gérer des exceptions. Pour prendre en charge ces exceptions, il faut partager beaucoup d'informations. Trouver la bonne information implique de trouver la ou les personnes qui la détiennent. Les réseaux sociaux d'entreprise viennent pallier ce manque autour des organisations traditionnelles" [**?**]. Les réseaux sociaux sont de plus en plus investis par les entreprises qui les utilisent comme outils marketing pour communiquer auprès des clients, pour mieux connaître et mieux définir leurs cibles, pour mieux cerner leur positionnement et la façon dont ils sont perçus, pour réaliser de la veille commerciale. Ces nouveaux outils ont créé de nombreux nouveaux concepts marketing tel que le marketing viral [20] , le marketing relationnel [21] , ou le marketing communautaire [22]. Comme exemple dans ce cas on peut citer *Linkedln*, *Viadeo*, *Xing*.

18. www.super-secretaire.com/magazine/missions-specialisees/gestion-rh/cid4891-reseau-social-d-entreprise-a-quoi-ca-sert.html, dernière visite le 16 février 2011

19. www.indexel.net/management/reseaux-sociaux-comment-en-tirer-profit.html, dernière visite le 19 février 2011

20. Le marketing viral : désigne toutes les techniques qui utilisent les outils du web en vue d'augmenter la notoriété d'une entreprise, ou de faire circuler les bruits ou les rumeurs (Ouvrage "Marketing, l'essentiel pour comprendre, décider, agir" par Marc Vandercammen). Le marketing viral utilise le potentiel de communication du consommateur, qui va communiquer de son côté, pour augmenter la notoriété d'un produit ou d'un service. Le consommateur contribue donc à la publicité du produit ou du service. Cela s'apparente en quelque sortes à du bouche à oreille sur internet.

21. Le marketing relationnel : désigne une technique de marketing centrée sur le client et sa fidélisation.

22. Le marketing communautaire : consiste à identifier des groupes d'affinité dans le but de leur proposer un message spécifique.

Chacun de ces réseaux, quelque soit la catégorie d'appartenance peut être soit ouvert, c'est-à-dire accessible à tout le monde, soit fermé, et donc uniquement accessible sur invitation. Comme tout outil, un réseau social présente un certain nombre d'intérêts et de limites.

Nous avons présenté une liste non exhaustive des caractéristiques des réseaux sociaux. De ces caractéristiques, il en découle plusieurs combinaisons possible. De ce fait, il est difficile de proposer une classification définitive des réseaux sociaux.

1.6 Intérêts des réseaux sociaux

Appartenir à un réseau social c'est adhérer à un facteur unificateur commun pour lequel on partage les mêmes principes et règles. Comme nous avons pu le remarquer, la caractéristique principale d'un réseau est l'interaction entre les membres.

Un premier intérêt des réseaux sociaux est la suspension des barrières de communication. Les réseaux sociaux rassemblent un très grand nombre de personnes. Une fois la communauté constituée, il devient très facile de faire de nouvelles connaissances, de partager et d'échanger des informations. Ceci s'illustre aisément lorsqu'on voit à quel point *facebook* a été l'un des principaux outils lors de la récente révolution tunisienne. Un second intérêt est qu'il permet à des personnes qui se sont perdues de vue de renouer les liens, ou d'établir de nouvelles relations. De plus , on constate que divers acteurs s'intéressent aux réseaux sociaux. C'est le cas par exemple des entreprises qui s'en servent pour le recrutement du personnel. "En tant que recruteur, nous utilisons quotidiennement le site Viadeo afin d'identifier des profils de candidats potentiels, travaillant ou ayant travaillé dans une entreprise donnée. Il suffit d'utiliser des mots-clés comme le nom de l'entreprise, ou de les associer à un domaine de compétence, à une formation ou une zone géographique. Pour obtenir les profils des candidats correspondants, c'est très pratique", explique Caroline Ledeuil, chargée de recherche au cabinet Altedia. D'autres entreprises l'utilisent pour promouvoir leurs produits, leurs marques. Nous notons aussi que le phénomène web 2.0 apporte des caractéristiques nouvelles. Nous remarquons par exemple que les utilisateurs peuvent désormais participer au développement de la plate forme, en ajoutant de nouveaux contenus, des groupes. Nous pouvons aussi citer une importation de conte-

nus multimédia tels que les vidéos, les images, les flux RSS [23].
Les réseaux sociaux comportent aussi quelques inconvénients. Certains utilisateurs peuvent s'y consacrer tellement et finir par créer un monde imaginé, en redéfinissant leur personnalité, leur identité et en renvoyant une image complètement fausse de ce qu'ils sont. Ce qui peut entraîner ces personnes à se replier sur elles mêmes.Un autre problème souvent rencontré est la divulgation d'informations personnelles voire intimes. La question est de savoir si la confidentialité des informations est belle et bien respectée par les personnes adéquates.
Notons aussi que le fait de vivre en communauté sous entend restrictions, barrières. De ce fait, ceux qui ne font pas partie d'une communauté peuvent se voir mettre à l'écart et se sentir marginalisés. Ceci implique une segmentation plus ou moins implicite de la population, car l'appartenance ou pas à une communauté peut être vue comme une prise de position face une situation donnée.
Avec les réseaux et les avancées technologiques, les informations sont échangées de manière spontanée. Cet atout qui fait la force des réseaux est aussi une faiblesse. Car il devient compliqué de canaliser les informations embarassantes. Dès lors, quelque soit l'entité agissant dans un réseau, et ce dans n'importe quel type de réseau, force est de constater que les acteurs ont tendance à se préoccuper de leur réputation au sein de la communauté. Si nous admettons que l'individu agit fortement sur son groupe social et que le groupe crée une contrainte qui pèse en retour sur les choix, les orientations, les comportements, les opinions des individus, alors il devient important de mieux analyser ces réseaux humains : leurs structures, leurs normes, et la position de chaque individu. Dans le cadre de notre travail,nous allons nous focaliser sur le réseau Twitter.

1.7 Etude de cas : Twitter

Twitter est à la fois un moyen de communication, un lieu de partage, de consommation d'information, un forum de discution et même un outil

23. RSS : Really Simple Syndication. Il s'agit concrètement d'un format simplifié pour exporter de l'information. Ces fichiers peuvent être lus par les navigateurs les plus récents ou par des lecteurs spécifiques dits "agrégateurs" (puisqu'ils vont vous permettre, à partir d'une sélection de flux qui vous intéressent, d'agréger et d'organiser des contenus), ou encore par des sites internet qui peuvent trouver un intérêt à rediffuser dans leurs pages un flux d'informations venant de l'extérieur. Cette dernière situation correspond à ce que l'on appelle la "syndication de contenus". Un flux RSS contient les dernières informations publiées par un site. C'est une sorte de "robinet" au quel on peut se brancher pour accéder rapidement et directement à un contenu.

d'écoute et d'analyse. Il peut aussi être vu comme un moteur de recherche en temps réel, ceci à cause de la grande quantité d'information qui y circule.

Twitter qui signifie "gazouillis" en anglais, est l'un des outils de réseaux sociaux les plus utilisés actuellement. Il permet à un utilisateur de décrire aux membres de son réseau "ce qu'il est entrain de faire". Il est possible d'envoyer et de recevoir ces notifications via internet, messagerie instantanée ou par messagerie numérique [24] .

En ce qui concerne l'utilisation, la procédure est semblable à celle des autre réseaux sociaux. Il suffit de s'enregistrer, une fois que c'est fait, rechercher et inviter des amis à se joindre à votre réseau. Chaque membre dispose de 140 caractères pour diffuser un message. Suivant le paramétrage effectué, un message généralement nommé *tweets* peut être accessible à tous, ou uniquement aux membres de votre réseau. En parallèle il y a possibilité de choisir des membres de Twitter dont ont veux suivre les publications. Outre cette concision imposée, Twitter se différencie des autres blog par le fait qu'il n'invite pas les lecteurs à commenter les messages postés.
Vu que nous serons appelé à extraire et à analyser les messages provenant de Twitter, il est intéressant de présenter quelques règles syntaxiques caracteristiques du vocabulaire sur Twitter :

- Un nom précédé d'arrobase est un lien vers le compte Twitter de l'utilisateur de ce nom. Il permet de voir tous les tweets de ce dernier, sauf s'ils sont protégés. Si le nom comprend un espace, il est remplacé par un tiret bas.
- Un mot précédé par "#" est un *mot-clic* encore appelé *hashtag*. Le hashtag est un moyen d'ajouter de l'information aux tweets pour les catégoriser suivant le contexte.
- On commence par "RT@Bob" un message de Bob que l'on relaie. Cette opération s'appelle un *retweet*.

Outre le site internet, l'accès à Twitter peut se faire via un desktop widget, ou encore via un agrégateur de flux RSS. Comme toute application très utilisée, Twitter est aussi présent sur les téléphones portables de nouvelle génération. Pour terminer, nous dirons que Twitter est un relai d'information à chaud. Ceci s'est matérialisé par le fait que plusieurs informations ont d'abord circulé sur Twitter avant d'être divulguées par les médias,et Laurent Suply [25] de déclarer dans le *Figaro* que "rien à ma connaissance ne va sur cette Terre

24. www.easy-concept.com/blog/ ?2009/03/16/743-twitter-outil-de-reseau-social-et-de-microblogage, dernière visite le 19 février 2011

25. Journaliste au Figaro.fr, il a une prédilection pour les thématiques scientifiques et technologiques

plus vite que Twitter. Ni moi, ni les télés, ni les agences".

Dans ce chapitre, nous avons présenté et abordé le concept de réseaux sociaux. Le chapitre suivant se basera sur les principes de la théorie des graphes pour proposer une représentation de ces réseaux. Nous présenterons les principes de cette théorie qui nous permettrons de mieux comprendre et de représenter les réseaux sociaux.

Chapitre 2

Analyse des réseaux sociaux : Formalisation et approche par la théorie des graphes

Un réseau social peut se définir comme étant un ensemble de relations entre un ensemble d'acteurs. Cet ensemble peut être organisé (c'est le cas dans une entreprise) ou non (c'est le cas d'un réseau d'amis). Ces relations peuvent être de natures diverses, spécialisées ou non, symétriques ou non [Lem99]. Des études ont été faites sur les interactions entre les entités, et des recherches pionnières ont été menées sur ces questions tant par des sociologues que par des ethnologues. Ces travaux sont à l'origine de l'important développement de l'analyse des réseaux sociaux auquel on assiste depuis le début des années 1970 [DF99].

L'analyse des réseaux sociaux est une approche sociologique fondée sur la théorie des réseaux sociaux. Elle vise à décrire les réseaux de relation, à traiter la description des schémas relationnels et à examiner comment la participation dans de tels réseaux sociaux contribue à expliquer le comportement et les attitudes des membres du réseau. Elle trouve ses fondements mathématiques dans la théorie des graphes.

La théorie des graphes fait partie des théories récentes en mathématiques. L'originalité de cette approche réside dans le fait que des points reliés par des lignes soient capables d'illustrer des situations mathématiques, sociologiques, géographiques, économiques, c'est-à-dire des situations diverses. En effet, les applications de la théorie des graphes fournissent des éléments mathématiques indispensables pour mieux approcher les réseaux[Bak93] . Cette représentation est une alternative pour la compréhension des réseaux

sociaux.

2.1 Représentation des réseaux sociaux

Selon Moreno, Jacob Levy est la première personne à avoir représenté un réseau social. Son objectif était d'avoir une vue graphique du réseau. Il a représenté les personnes par des points et une relation entre deux personnes était représentée par une flèche [EGBG09]. Cette représentation est depuis désignée par le terme *sociogramme*. Mais le terme *toile* était aussi utilisé ; ceci en référence à l'aspect de toile d'araignée souvent observée.

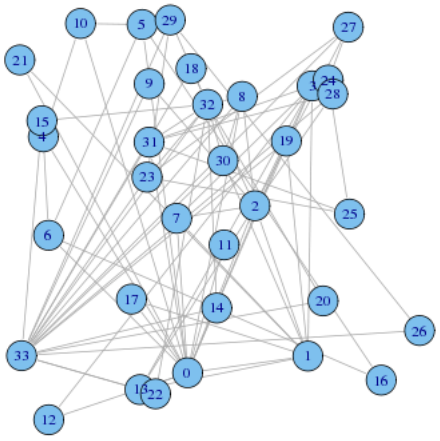


FIGURE 2.1 – Exemple de réseau social : le club de karaté de Zachary

La figure 2.1 représente le réseau social d'un club de karaté de 34 amis d'une université aux Etats Unis en 1970. Il a été construit par Zachary [Zac77]. Sachant qu'un sommet est l'unité de base d'un réseau et une arête une connexion entre deux sommets, les mathématiciens ont très vite fait le rapprochement entre le sociogramme et la théorie des graphes. C'est ainsi que Cartwright et Harary [EGBG09] ont appliqué la théorie des graphes à l'analyse des réseaux sociaux. La théorie des graphes est ainsi devenue le principal outil de représentation des réseaux sociaux. Elle dispose de théorèmes, d'algorithmes et de raisonnements propices à l'analyse des réseaux. Cette

théorie est bien plus qu'un outil de représentation graphique, c'est un outil de conception. Il est donc important de rappeler quelques concepts de base de la théorie des graphes.

2.1.1 Les sommets

Encore appelés nœuds, entités, éléments, les sommets peuvent représenter plusieurs choses. Ils représentent parfois les individus ou des structures sociales telles que des groupes, des équipes, des organisations, des institutions, des états et même des pays. Dans certaines circonstances, les sommets peuvent représenter des contenus tels que les pages web, les mots clés ou des vidéos. Parfois, on associe à ces sommets des attributs qui permettent de les caractériser. Ces attributs peuvent être utilisés lors de l'analyse du réseau. C'est le cas par exemple dans Twitter lorsqu'à un individu on lui associe le nombre de personnes qui le suivent ou qu'il suit.

2.1.2 Les liens

Aussi connus sous les termes relations et connexions, ils permettent de représenter l'existence d'une liaison entre deux sommets quelconques. Ces liens permettent de matérialiser différents types de liaisons telles que les relations de proximité, de collaboration, de compétition, d'amitié, de parenté et même de partenariat. Un lien est donc toute forme de relation ou de connexion entre deux entités.

Ces liens peuvent être orientés ou non. Lorsqu'ils sont orientés, on parle d'*arc* ; dans ce cas le graphe est dit *orienté*, on parle de *digraphe* . Dans le cas contraire on parle d'*arêtes*, ce qui correspond au graphe *non orienté*. Les graphes non orientés sont adaptés pour l'analyse des des réseaux sociaux dans lesquels l'interaction entre deux entités en relation se fait dans les deux sens, c'est-à-dire dans le cas d'une relation symétrique. Les graphes orientés quant à eux ont une grande utilité dans la représentation des relations non symétriques, unidirectionnelles. C'est le cas pour les réseaux de confiance [1] par exemple. Dans un réseau social, le fait qu'un utilisateur Twitter suit un autre peut être matérialisé par un arc. Dans Facebook par exemple nous pouvons schématiser la relation d'amitié entre deux utilisateur par une arête.
Ces liens peuvent aussi être caractérisés par des poids. Dans ce cas, on a à faire à des liens *pondérés*. Dans le cas contraire, ils sont *non pondérés*. Par

1. Un réseau de confiance est une sorte de modèle de confiance,c'est un réseau tissé autour de soi avec des personnes que l'on connaît,et avec qui on se fait des échanges de clés par exemple. Il est utilisé dans le domaine de la cryptographie. [INF]

exemple, dans Facebook soit deux utilisateurs sont amis soit ils ne le sont pas. Cette relation d'amitié sera représentée par une arête non pondérée. Cependant, une arête pondérée permettra de représenter le nombre de messages que deux utilisateurs facebook se sont échangés.

2.1.3 Concepts de base de la théorie des graphes

Traditionnellement, un *graphe* G se définit comme une paire d'ensembles. $G = (S, L)$, où S représente l'ensemble des sommets, et L l'ensemble des liens entre les sommets. Nous définissons ci-dessous quelques mesures et indices usuels de la théorie des graphes.

- Le nombre de sommets $|S| = n$ dénote l' ***ordre*** du graphe. Dans un graphe, l'ordre minimum est de 0. Dans ce cas le graphe est dit ***vide***.

- Le nombre de liens $|L|$ d'un graphe représente la ***taille*** de ce graphe. La taille maximale est de $n(n-1)/2$. Dans ce cas, le graphe est dit ***complet***.

- La ***densité*** d'un graphe quant à elle est la proportion des liens existant par rapport aux liens possibles entre sommets. Elle indique la quantité de liens au sein d'un réseau et permet de définir la cohésion de ce dernier. La densité varie également en fonction du type de relations considérés dans un réseau. Un réseau basé sur des relations amoureuses est beaucoup moins dense qu'un réseau de relations d'amitiés, notamment en raison des caractéristiques des liens.

La figure 2.2 nous propose une représentation du graphe défini par les ensembles $S = \{1, 2, 3, 4, 5\}$ et $L = \{(1, 2), (1, 3), (2, 3), (3, 5), (3, 4)\}$. Pour pour la figure 2.2, nous avons un graphe de taille 5 et d'ordre 5.

- Dans un graphe, un ***chemin*** représente une séquence d'arcs qui relient deux nœuds donnés, alors qu'une ***géodésique*** définit l'un des plus courts chemin entre deux sommets donnés. Dans le cas des graphes non orientés, on parle de ***chaîne***.

- La ***distance*** entre deux sommets désigne la longueur du chemin qui les relie.

- Le ***degré d'un sommet*** est les nombre d'arêtes qui lui sont incidentes.

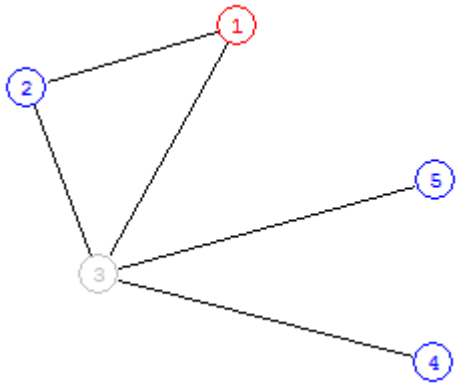


FIGURE 2.2 – Exemple de Graphe non orienté

- Le ***degré d'un graphe*** est le degré maximum de tous ses sommets.

- Un sommet est ***incident*** à une arête s'il est situé à une des deux extrémités de cette arête. Inversément, une arête est incidente à un sommet si elle "touche" ce sommet.

- ***L'indice de centralité du graphe*** se définit par la somme des indices de centralité de tous les sommets qui composent le graphe. La problématique de centralité d'un réseau social est de définir ce qui rend le nœud plus central qu'un autre, on parle de ***centralité locale***.

- La ***centralité de degré*** [Nie74] considère comme centraux les nœuds qui possèdent les degrés les plus élevés du graphe. Par leur forte connectivité aux autres éléments, ces nœuds ont un potentiel élevé à faire circuler l'information.

- La ***centralité d'intermédiarité*** qui se concentre sur la capacité d'un nœud à servir d'intermédiaire dans un graphe. Un nœud situé sur une géodésique possède une position stratégique dans la cohésion d'un réseau et dans la circulation de l'information, d'autant plus si ce chemin est unique.

- La ***centralité de proximité***, qui mesure la centralité d'un nœud en se basant sur la taille des chemins qui le lient aux autres nœuds. Cette mesure représente la capacité d'un nœud à se connecter rapidement avec les autres nœuds du réseau [Fre79].

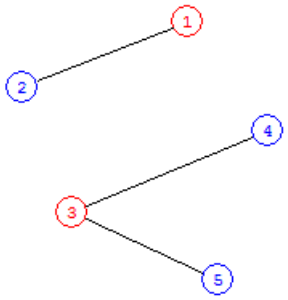


FIGURE 2.3 – Exemple de Graphe partiel déduit de la figure 2.3

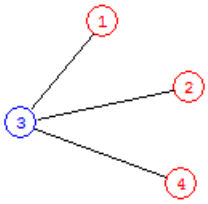


FIGURE 2.4 – Exemple de Sous graphe déduit de la figure 2.3

- La ***centralité égocentrée***, détermine l'influence d'un nœud par rapport à son voisinage.

- Dans certains cas, deux sommets d'un même graphe peuvent être liés par plusieurs arêtes ; ces arêtes sont dit ***parallèles***. De même, une arête peut avoir ses extrémités confondues : on parle de ***boucle***. Dans ces deux cas, le graphe considéré est un ***multigraphe***.

Soit $G = (S, L)$ un graphe. Si un ensemble L' est inclus dans L ,alors le graphe $G' = (S, L')$ qui en dérive est un *graphe partiel* de G. Autrement dit, on obtient G' en enlevant une ou plusieurs arêtes au graphe G (figure 2.3).

- Pour un sous ensemble de sommets A inclus dans S, le ***sous-graphe*** de G induit par A est le graphe $H = (A, L(A))$ dont l'ensemble des sommets est A et l'ensemble des arêtes $L(A)$ est formé de toutes les arêtes de G ayant leurs deux extrémités dans A. C'est-à-dire que H est obtenu en enlevant un ou plusieurs sommets au graphe G, ainsi que toutes les arêtes incidentes à ces sommets(figure 2.4).
- Un graphe partiel d'un sous-graphe est un ***sous-graphe partiel*** de G.

- Une ***clique*** un sous-graphe complet de G. Rappelons que dans un graphe complet, toutes les paires de sommets sont adjacentes ; deux sommets sont adjacents s'ils sont reliés par une arête.

- Le ***degré d'un sommet*** s, est le nombre d'arêtes incidentes à ce sommet.

- Le ***degré d'un graphe*** est le degré maximum de tous ses sommets. Dès lors, un graphe dont tous les sommets ont le même degré est dit ***régulier*** ; si le degré commun est par exemple k, on dit du graphe qu'il est ***k-régulier***.

- Le ***diamètre*** d'un graphe dénote la plus grande des distances entre deux sommets de ce graphe. Dans le domaine des réseaux sociaux, cette notion se rapproche de celle du petit monde proposée par Milgram[Mal09].

- Il existe une distinction entre les ***graphes pondérés*** et les ***graphes non pondérés***. Les graphes pondérés sont munis d'arêtes pondérées, c'est-à-dire d'arêtes auxquelles on attribue des poids, des valeurs. Les graphes pondérés sont adaptés aux réseaux sociaux qui contiennent plusieurs niveaux d'intensités dans les relations.

- Les graphes étiquetés quant à eux sont utiles pour la représentation de différents types de relations. Les graphes étiquetés sont des graphes avec des arêtes étiquetées. En ce qui concerne les ***graphes multipartites***, ils sont adaptés pour des réseaux sociaux incluant différents types de ressources manipulées par les acteurs et qui sont le support d'interactions. Ce sont des graphes avec des sommets de différents types.

La théorie des graphes propose aussi une représentation matricielle des graphes. La matrice est l'objet mathématique le plus utilisé pour manipuler ces relations. Notons aussi qu'il existe des approches ensemblistes qui avaient été proposées [Sco00] .

En théorie des graphes, on distingue les matrices d'incidence des matrices d'adjacence. Une matrice qui possède les mêmes entités en ligne et en colonne est une matrice d'*adjacence*. Une telle matrice est donc une matrice carrée. Sous cette hypothèse, un graphe peut se représenter sous forme d'une matrice M à n lignes et n colonnes représentant un tableau. Notons a_{ij} chaque élément correspondant à la ligne i et la colonne j de ce tableau. La valeur de l'élément a_{ij} constitue le poids de la relation entre l'entité i et l'entité

j. Dans un graphe non pondéré, a_{ij} vaut 1 s'il existe une relation entre les entités i et j ; la valeur 0 de a_{ij} correspond à une absence de relation entre les entités correspondantes.

Une matrice d'***incidence*** quant à elle représente une relation entre deux ensembles distincts d'entités. En ligne on a les éléments du premier ensemble, en colonne ce sont les éléments du second ensemble.

Remarquons qu'une matrice d'incidence peut se convertir en deux matrices d'adjacence représentant chacune les entités des lignes et des colonnes, les valeurs des cases contenant les points communs entre les entités correspondantes dans la matrice d'incidence, a_{ii} n'ayant pas de valeur. Nous avons un exemple (table 2.1) de matrice d'incidence indiquant sur quel projet travail un employé.

	Projet1	Projet2	Projet3	Projet4
Employé1	1	1	1	0
Employé2	1	0	0	0
Employé3	1	1	1	1
Employé4	0	0	1	1

TABLE 2.1 – Matrice d'incidence Employé/Projet

La table 2.2, nous montre la matrice d'adjacence déduite de la matrice d'incidence ci-dessus. Dans cette matrice, chaque case représente le nombre de projets partagés entre les employés correspondants.

	Employé1	Employé2	Employé3	Employé4
Employé1	-	1	3	1
Employé2	1	-	1	0
Employé3	3	1	-	2
Employé4	1	0	2	-

TABLE 2.2 – Matrice d'adjacence nombre de projets partagés entre employés

Une seconde matrice d'adjacence que nous pouvons déduire de la matrice d'incidence est celle dans laquelle chaque case représente le nombre d'employés repartis entre les projets correspondants.

	Projet1	Projet2	Projet3	Projet4
Projet1	-	2	2	1
Projet2	2	-	2	1
Projet3	2	2	-	2
Projet4	1	1	2	-

TABLE 2.3 – Matrice d'adjacence nombre d'employés par projet

Un graphe peut également se représenté par une matrice de *Laplace*. Les éléments a_{ij} de cette matrice se définissent comme suit [EGBG09] :

$$a_{ij} = \begin{cases} k_i & si \quad i = j \\ -1 & si \quad i \neq jet(v_i, v_j) \in E \\ 0 & autrement \end{cases}$$

k_i désigne le degré du nœud v_i

Dans cette section, nous avons présenté différentes formalisations des réseaux sociaux par l'approche de la théorie des graphes. Pour une meilleure analyse structurelle des réseaux, nous allons nous pencher sur la détection automatique des différents groupements qui peuvent se constituer à l'intérieur d'un réseau.

2.2 Détection de communautés

Il peut arriver qu'un graphe soit formé de n*oe*uds fortement connectés entre eux et faiblement avec les autres n*oe*uds du graphe. Ce sous-ensemble constitue ce qu'on appelle ***communauté***. De manière générale, une communauté décrit un ensemble d'individus ayant des caractéristiques communes.

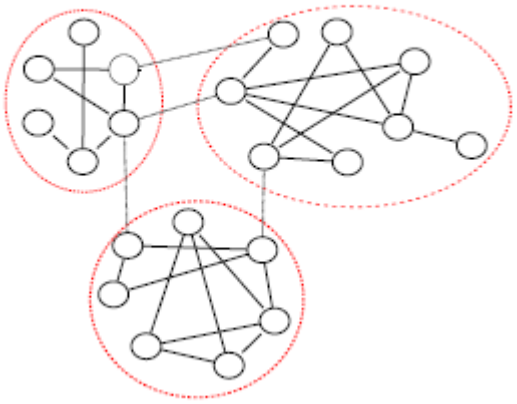


FIGURE 2.5 – Exemple de communautés

Dans un réseau, les communautés symbolisent des zones denses, où l'on rencontre une forte connexion entre les individus. Dans certains cas, ces communautés elles mêmes sont susceptibles d'en contenir d'autres, et ainsi à plusieurs niveaux de profondeur.
La notion de communauté intervient dans plusieurs domaines. On peut par exemple imaginer une communauté de toutes les relations d'une personne, elle même découpée en plusieurs groupements : un cercle familial, un cercle d'amis et un cercle de collègues. Le cercle familial peut encore être subdivisé en deux. Un sous groupe pour la famille maternelle, et un autre pour la famille partenelle.
Dans les réseaux sociaux, des structures communautaires ont pu être définies et observées. Il est alors intéressant d'essayer de détecter automatiquement ces communautés. Celles-ci ont un intérêt pour la compréhension de la structure du réseau, l'amélioration de la visualisation, et dans certains cas l'amélioration des moteurs de recherche. La détection de communautés permet d'identifier différents groupes dans un réseau social. Détecter une communauté signifie partitionner l'ensemble des nœuds du graphe.
Le terme *graph clustering* plutôt que *détection de communautés* est parfois employé dans la littérature. Ce problème peut être vu comme un jeune héritier des problèmes de clustering en data-mining et du problème de partitionnement de graphe en informatique. Quelque soit le domaine, les communautés recherchées semblent être relativement similaires. La découpe en

communautés consiste alors à découper et à identifier, au sein d'un réseau, les entités fortement connectées pour les regrouper. Elle permet de regrouper les nœuds fortement connectés entre eux et de séparer les nœuds peu connectés. De manière générale, le partitionement de graphe et le clustering de données sont les thèmes les plus classiques quand on parle de la détection de communautés [Pon07] . Ces approches se basent souvent sur les mesures de calcul de distances entre les sommets du graphe et les mesures de similarité. La notion de distance doit respecter certains critères. Soit d, la distance entre deux sommets. Etant donné les sommets x,y,z, la distance vérifie les propriétés suivantes :

- Séparation : $d(x, y) = 0$,
- Symétrie : $d(x, y) = d(y, x)$,
- Inégalité triangulaire : $d(x, z) \leq d(x, y) + d(y, z)$.

Etant donné les sommets x,y,z, la mesure de similarité s quant à elle vérifie les critères suivants :

- $s(x, x) = k$ où k est une constant,
- $s(x, y) = s(y, x)$,
- $s(x, y) \leq s(x, x) = k$.

Notons qu'il est possible de transformer une distance en mesure de similarité. Comme méthodes traditionnelles nous pouvons citer les méthodes de clustering hiérarchique, les méthodes spectrales et les méthodes de partitionnement.

2.2.1 Les méthodes de partitonnement

Le partitionnement a pour but de regrouper les sommets d'un graphe en un nombre prédéfini (k) de parties, en minimisant le nombre le liens entre les différents groupes. Cette approche présente une importante limite pour la détection des communautés : il faut connaître dès le départ le nombre de communautés recherchées ainsi que leurs tailles, ce qui est souvent très difficile à prédire dans le cadre des réseaux sociaux.
Une méthode populaire et très utilisée est le *k-means*, car simple et rapide. Au départ de l'algorithme, on choisit k sommets formant ainsi k clusters. Ensuite, on affecte chaque sommet s au cluster C_i de centre M_i tel que la distance entre le sommet s et le centre M_i soit minimale. le centre du groupe est la moyenne de tous les points de ce groupe. En plus du fait qu'il faut choisir un nombre de cluster dès le départ, l'algorithme du *k-means* présente une sensibilité à la partition initiale, ce qui peut entraîner la convergence vers un optimum local et donc non global.

2.2.2 Les méthodes hiérarchiques

Suivant la manière avec laquelle se fera le partitionnement, on divise les methodes hiérarchiques en deux types : les méthodes agglomératives et les méthodes divisives[Pon07]. Nous allons définir chacune de ces méthodes et présenter de manière succinte quelques algorithmes associés.

L'approche agglomérative

L'idée de cette approche est que les sommets sont regroupés itérativement en communauté en partant d'une partition où chaque communauté est composée d'un seul sommet. L'arrêt se fait lorsqu'on on obtient une communauté regroupant tous les sommets. Une structure hiérarchique appelée *dendrogramme* est ainsi construite. Pour la fusion des communautés, on utilise la notion de mesure de dissimilarité D.
Etant donné A et B deux clusters, et x et y deux sommets distincts, il existe plusiuers façons de définir la distance D entre communautés :

- *Single linkage* : c'est la distance la plus simple. Elle considère que la distance entre deux communautés est la distance minimale entre deux sommets de celles-ci.
$$D(A,B) = min\{d(x,y) : x \in A, y \in B\}$$
- *Complete linkage* : cette approche considère la plus petite distance maximale entre deux communautés.
$$D(A,B) = max\{d(x,y) : x \in A, y \in B\}$$
- *Average linkage* : dans ce cas, la distance entre deux communautés est la moyenne des distances entre chaque paire de leurs sommets.
$$D(A,B) = \frac{\sum_{x \in A} \sum_{y \in B} d(x,y)}{|A|.|B|}$$

Quelques algorithmes de cette approche :

- **L'algorithme de Donetti et Muñoz** [DM05] : cet algorithme utilise les propriétés spectrales de la matrice Laplacienne du graphe pour détecter les communautés. Les coordonnées i et j des vecteurs propres correspondant aux plus petites valeurs propres non nulles sont corrélées lorsque les sommets i et j sont dans la même communauté.
- **L'algorithme de Clauset et al** [ACM04] : il se base sur la notion de modularité introduite par Newman. Cette notion se fonde sur les proportions d'arcs internes aux communautés et les proportions d'arcs liés à chaque communauté. Seules les communatés ayant un arc entre elles peuvent être fusionnées à chaque étape. Une optimisation de cette méthode a été recement proposée par Wakita et Tsurumi [WT07].

L'approche divisive

Cette approche essaye de scinder le graphe en plusieurs communautés en retirant progressivement les liens entre les différentes communautés. A chaque étape, les composantes connexes sont identifées comme des communautés, et les liens entre ces communautés sont retirés. On obtient, tout comme dans le cas de l'approche agglomérative une structure hiérarchique. Les algorithmes proposés dans ce cadre diffèrent par la façon de sélectionner les liens à supprimer. Quelques algorithmes utilisant cette aproche :

- **L'algorithme de Girvan et Newman** [NG04] : dans cet algortime, on retire les arcs de plus forte centralité d'intermédiarité. Pour un arc, cette centralité se définie comme le nombre de plus court chemin passant par lui.
- **L'algorithme de Fortunato et al** [SFM04] : c'est une variété de l'algorithme de Girvan et Newman. Il est basé sur le centralité d'information. Ici, les auteurs ont défini l'efficacité de communication entre deux sommets comme étant l'inverse de leur distance. L'efficacité du réseau est alors défini comme la moyenne de l'efficacité de tous les couples de sommets. Cette approche donne de meilleurs résultats que celle de Girvan et Newman.
- **L'algorithme de Duch et Arenas** [DA05] : Cette méthode consiste à diviser le graphe en deux communautés et à diviser récursivement les deux communautés ainsi trouvées. Chaque étape part d'une division arbitraire et des sommets sont ensuite changés de communauté de façon à améliorer la modularité. Le sommet à déplacer est choisi aléatoirement parmi les sommets ayant les moins bonnes contributions à la modularité globale de la coupe. La coupe obtenant la meilleure modularité est retenue tout au long du processus.

La méthode de clustering spectrale quant à elle est basée sur les vecteurs propres. Elle consiste à extraire les vecteurs propres associés aux valeurs propres de la matrice qui représente le graphe.

Le problème de la classification est traité dans plusieurs communautés de recherche.Les méthodes de classification supervisées peuvent aussi être intéréssantes dans la détection des communautés. Plusieurs méthodes de classification supervisée publiées dans la littérature s'appuient sur des techniques différentes. Parmi celles ci, nous avons la méthode basée sur les treillis de concepts. L'intérêt du treillis est qu'il permet de restituer entièrement le concept décrit par les données.

"Les treillis de concepts formels [2] sont des structures mathématique permettant de représenter les classes non disjointes sous-jacentes à un ensemble d'objets (exemples, instances, tuples ou observations) décrits à partir d'un ensemble d'attributs (propriétés, descripteurs ou items). Ces classes non disjointes sont aussi appelées concepts formels, hyper-rectangles ou ensembles fermés. Une classe matérialise un concept, une idée générale que l'on a d'un objet" [NJI05] .
Cette approche par les treillis de Galois, peut être affinée de sorte à mettre en évidence les concepts fondamentaux de l'algèbre linéaire : Ordre et Treillis.

2. Les treillis de concepts formels ou treillis de Galois

Chapitre 3

Analyse formelle de concepts

Les méthodes présentées dans le chapitre précédent génèrent des résultats de manière non exhaustive, à chaque exécution, elles produisent un résultat différent. L'analyse formelle de concepts et les treillis de Galois sont une alternative à ce problème. L'analyse formelle de concepts est une application de la théorie des treillis basée sur la formalisation de la notion de concepts et de regroupement conceptuel. Elle permet, entre autres, la construction du treillis de concepts. La construction du treillis s'est avérée être un cadre théorique intéressant pour la fouille de données puisqu'il permet la génération de concepts.
L'analyse formelle de concepts est un domaine de recherche très large. Dans ce chapitre, nous allons nous concentrer essentiellement sur les éléments qui permettent de définir et de construire un treillis de concepts.
Tout d'abord, nous présenterons les généralités sur les notions d'ordres et de treillis. Ensuite, l'Analyse Formelle de Concepts (AFC) sera présentée.

3.1 Ordre et Treillis

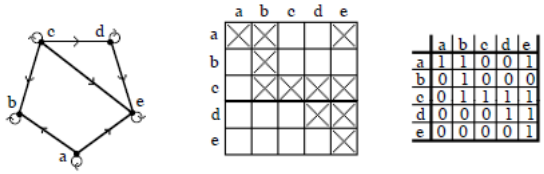
Nous présentons ici les notions mathématiques nécessaires à la définition des Treillis.

3.1.1 Relation binaire

Une **relation binaire** R sur un ensemble X est un ensemble de couples de X, c'est un sous ensemble de X x X.

Exemple : $X = \{a, b, c, d, e\}$
$R = \{(a, b), (a, e), (c, b), (c, d), (c, e), (d, e), (a, a), (b, b), (c, c), (d, d), (e, e)\}$

La figure 3.1, propose 3 façons de représenter cette relation.



	a	b	c	d	e
a	1	1	0	0	1
b	0	1	0	0	0
c	0	1	1	1	1
d	0	0	0	1	1
e	0	0	0	0	1

FIGURE 3.1 – 3 représentations de la relation d'ordre

3.1.2 Relation d'ordre

La relation R est une **relation d'ordre** si elle est :
- *réflexive* : pour tout $x \in X$, xRx,
- *antisymétrique* : pour tous $x, y \in X, xRy$ et $yRx \Rightarrow x = y$,
- *transitive* : pour tous $x, y, z \in X, xRy$ et $yRz \Rightarrow xRz$.

De manière usuelle, la relation d'ordre se note $\leq$

Soit $\leq$ une relation d'ordre sur X. On peut définir la **relation d'ordre inverse** $\geq$ de la manière suivante. Si x et y sont deux éléments de $X, x \geq y$ si et seulement si $y \leq x$.
Par exemple, sur l'ensemble des entiers, on a la relation *supérieure ou égale* ($\geq$) qui est la relation inverse de la relation *inférieure ou égale* ($\leq$).

La relation R est une **relation d'ordre stricte** si elle est :
- *irréflexive* : pour tout $x \in X, xRx$,
- *transitive* : pour tous $x, y, z \in X, xRy$ et $yRz \Rightarrow xRz$
  irréflexive + transitive $\Rightarrow$ *asymétrique* : pour tous $x, y \in X, xRy \Rightarrow yRx$

De manière usuelle la relation d'ordre stricte se représente par $<$.

3.1.3 Ensemble ordonné

Un **ensemble ordonné** est un couple $(X, \leq)$ où X est un ensemble et $\leq$ une relation d'ordre sur X.
Exemple : $(\mathbb{N}, <)$ est ordonné.
Tout ensemble ordonné peut être représenté par un diagramme appelé **diagramme de Hasse** ou **diagramme de couverture**.
Soit $(X, \leq)$ un ensemble ordonné. Un diagramme de Hasse est une représentation graphique (N, A) où N est un ensemble de nœuds représentants les éléments

de X (donc $|N| = |X|$), et A un ensemble d'arêtes. Il y a une arête entre deux nœuds n_1 et n_2 si et seulement si n_2 couvre n_1. x est *couvert* par y (noté $\prec$)s'il n'existe pas d'élément "entre" x et y : $x \prec y$ si $x < y$ et $x \leq z < y \Rightarrow x = z$. Si $x, y \in X$ et $x \prec y$, alors le cercle représentant le sommet y doit être au-dessus de celui représentant x. Les deux cercles étant reliés par un segment. Pour illustrer le diagramme de Hasse, considérons l'ensemble $X = \{60, 30, 20, 15, 12, 10, 6, 5, 4, 3, 2, 1\}$, et la relation considéré est la relation de divisibilité.

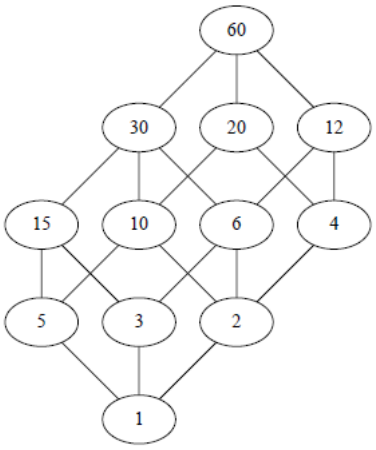


FIGURE 3.2 – Diagramme de Hasse

A partir de ce diagramme, on peut déduire que $x \prec y$ si et seulement si il existe un chemin ascendant qui relie le cercle correspondant à x à celui correspondant à y.
Soit $(X, \leq)$ un ensemble ordonné, et S un sous-ensemble non-vide de X.

On dit que l'élément $x \in X$ est un ***majorant*** de S si et seulement si $\forall y \in S, y \leq x$.

On dit que l'élément $x \in X$ est un ***minorant*** de S si et seulement si $\forall y \in S, x \leq y$.

En considérant l'exemple ci-dessus, si on prend $S = \{20, 30, 6\}$, les minorants sont $\{2, 1\}$ et le seul majorant est $\{60\}$.
Soit $(X, \leq)$ un ensemble ordonné, et S un sous-ensemble non-vide de X.
L' *infimum* noté $\bigwedge$ est le plus grand des minorants de S.
Le
textbftextitsupremum noté $\bigvee$ est le plus petit des majorants de S.

Si on reprend l'exemple précédent où $S = \{20, 30, 6\}$, l'infimum est 2 (le plus grand élément de l'ensemble $\{2, 1\}$ des minorants) et le supremum est 60.

3.1.4 Les treillis

Un ensemble ordonné $(T, \leq)$ est un **treillis** si et seulement si tout couple d'éléments (x, y) de T possède une borne supérieure unique (supremum : SUP) noté $x \vee y$, et une borne inférieure unique (infimum : INF) notée $x \wedge y$. La figure 3.3 nous donne un exemple de treillis construit à partir de l'ensemble $E = \{a, b, c\}$ et de la relation d'inclusion. Soit S l'ensemble des éléments de ce treillis : $S = \{\{\}, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

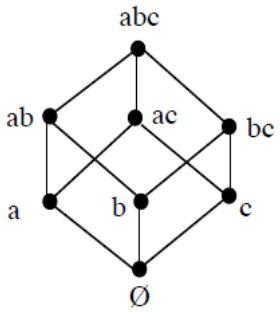


FIGURE 3.3 – Exemple de treillis

Soient x, y et z les élements d'un treillis. Les opérations $\vee$ et $\wedge$ possèdent les propriétés suivantes :

- *commutativité* : $x \vee y = y \vee x$ et $x \wedge y = y \wedge x$,
- *associativité* : $x \vee (y \vee z) = (x \vee y) \vee z$ et $x \wedge (y \wedge z) = (x \wedge y) \wedge z$,
- *idempotence* : $x \vee x = x$ et $x \wedge x = x$,
- *absorption* : $x \vee (y \wedge z) = x$ et $x \wedge (y \vee z) = x$,
- $x < leqy$ ssi $x \vee y \leq= x$ ou $x \wedge y = y$,
- *monotonie* : si $y \leq z$, alors $x \vee y \leq x \vee z$ et $x \wedge y \leq x \wedge z$.

Un treillis est ***modulaire*** si pour $x \leq z$ alors $x \vee (y \wedge z) = (x \vee y) \wedge z$.

Un ***treillis distributif*** si $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

Un ***treillis booléen*** est un treillis distributif où chaque élement a un complément.

Etant donné un treillis $(X, \leq)$ et un élément x de X, un complément de x est un élément $y \in X$ tel que $x \vee y = \bot$ et $x \wedge y = \top$.

3.1.5 Les treillis de concepts

Contexte formel

Un ***contexte formel*** K est un triplet de la forme (O, A, I) où O représente un ensemble d'objets, A un ensemble d'attributs décrivant les objets, et I une relation entre l'ensemble des objets et l'ensemble des attributs ; $I \subseteq OxA$. Le couple $(o, a) \in I$ avec $o \in O$ et $a \in A$ indique que l'objet o possède l'attribut a ou encore que l'attribut a est possedé par l'objet o.
Dans la pratique, beaucoup de problèmes se modélisent sous forme de contextes formels car les données associées à ce problème peuvent être ediscrétisables pour ensuite former une relation binaire mettant en liaison des objets et des attributs. L'étape de prétraitement des données généralement nécessaire consiste à adapter un problème pour le transformer sous forme de contexte qui a du sens et à partir duquel on peut extraire de l'information.
Un contexte peut être représenté graphiquement par un tableau de dimension $|O|$x $|A|$. Il est donc possible d'avoir une approche objet (en ligne) ou bien attribut (en colonne) selon les besoins. Le résultat est toujours le même quelque soit l'approche. On peut voir un exemple de contexte formel dans le tableau 3.1 qui propose une description de certains animaux.

	aDesPoils	estOvipare	aDesDents	aDesNageoires	aDesPlumes
Antilope	X		X		
Sanglier	X		X		
Poulet		X			X
Poisson chat		X	X	X	

TABLE 3.1 – Table de contexte

Pour chaque ensemble d'objets $X \in P(O)$[1], les attributs partagés par ces objets peuvent être obtenus à l'aide de l'application f. Cette application est appelée ***extension***.

$$f : P(O) \rightarrow P(A),\ f(X) = \{\ a \in A | \forall\ o \in X,\ (o,a) \in I\ \}$$

1. $P(X)$ représente l'ensemble des parties de X

Symétriquement, l'application g associe à un ensemble d'attributs tous les objets partagés. L'application g se nomme ***intension***.

$$g : P(A) \rightarrow P(O),\ g(Y) = \{\ o \in O | \forall\ a \in Y,\ (o,a) \in I\ \}$$

Par exemple sur la table de contexte 3.1 on a :
f(Antilope,Sanglier) = { *aDesPoils, aDesDents*}
g(aDesPlumes) = { *Poulet*}

Contexte binaire

Un contexte *(O,A,I)* est dit ***binaire*** si les élements de A ne peuvent prendre que deux valeurs (0 ou 1) qui indiquent l'absence ou la présence de l'attribut concerné dans la description de l'objet.
Le contexte binaire correspondant au contexte formel de la table 3.1 se présente comme suit :

	aDesPoils	estOvipare	aDesDents	aDesNageoires	aDesPlumes
Antilope	1	0	1	0	0
Sanglier	1	0	1	0	0
Poulet	0	1	0	0	1
Poisson chat	0	1	1	1	0

TABLE 3.2 – Table de contexte binaire du contexte de la table 3.1

Fermeture des ensembles

Les deux fonctions f et g vont servir à calculer la fermeture de X et Y représentant respectivement un sous-ensemble d'objets et un sous-ensemble d'attributs. Pour cela, on compose les fonctions f et g comme suit :

$$X^{"} = g(f(X))$$
$$Y^{"} = f(g(Y))$$

On dit qu'un ensemble est ***fermé*** s'il est égal à sa fermetue. Ainsi, X est fermé si $X = X^{"}$ et Y est fermé si $Y = Y^{"}$.
Dans la pratique, le calcul de la fermeture est très coûteux puisqu'il nécessite de calculer g et f. Le calcul d'une seule correspondance nécessite de parcourir

tout le contexte qui peut être de très grande taille. Or la génération d'un treillis de concept se fait sur base d'ensembles fermés. Les algorithmes de génération de treillis de concepts se différencieront surtout sur leurs façons de minimiser le calcul des fermetures.

Concept

Un ***concept formel*** d'un contexte $K=(O,A,I)$ est une paire (X,Y) avec $X \in P(O)$, $Y \in P(A)$ tels que $f(X) = Y$ et $g(Y) = X$.
$f(X)$ est l'ensemble de tous les attributs de Y possédés par les objets de X. De façon duale, $g(Y)$ est l'ensemble de tous les objets possédant les attributs de Y.

Concepts	**({Extension, Intension})**
0	({Antilope, Sanglier, Poulet, PoissonChat},∅)
1	({Antilope, Sanglier},{aDesPoils})
2	({Antilope, Sanglier, PoissonChat},{aDesDents})
3	({PoissonChat},{estOvipare})
4	({Antilope, Sanglier},{aDesPoils, aDesDents})
5	({PoissonChat},{aDesPoils, aDesNageoires, estOvipare})
6	({Poulet},{aDesPlumes, estOvipare})
7	(∅,{aDesPoils, estOvipare, aDesDents, aDesNageoires, aDesPlumes})

TABLE 3.3 – Concepts du contexte de la table 3.1

L'ensemble des concepts extrait d'un contexte formel peut être ordonné dans un treillis.

Treillis de concepts

L'ensemble C des concepts extrait d'un contexte formel muni de la relation d'ordre partiel $\leq_L$ forme un treillis $L = (C, \leq_L)$. La relation d'ordre partiel $\leq_L$ entre les concepts correspond à l'inclusion des extensions (ou l'inclusion inverse des intensions).

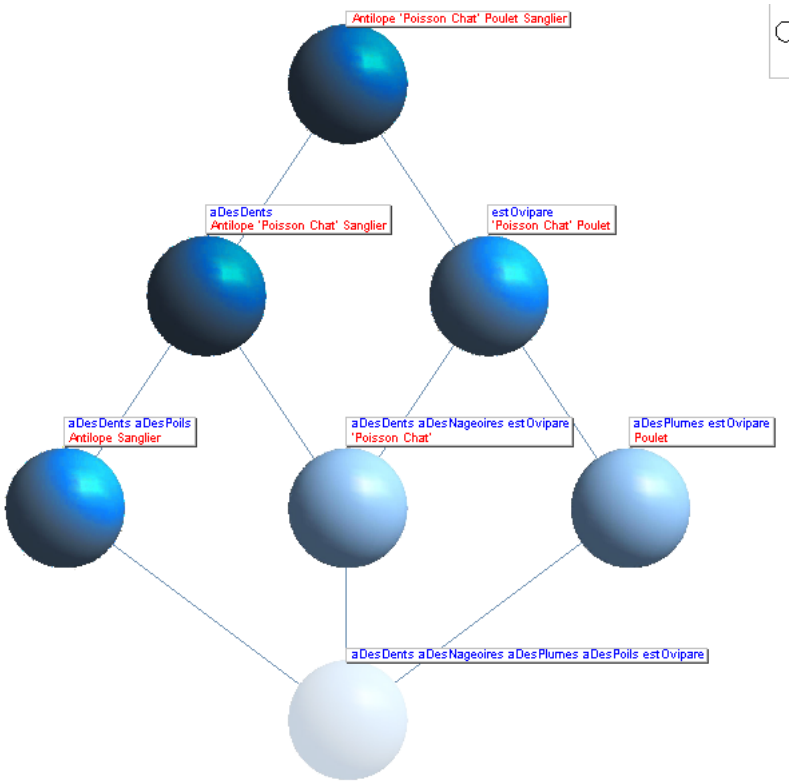


FIGURE 3.4 – Treillis du contexte de la table 3.1

Dans la figure 3.1, on peut voir que
$(\{PoissonChat, Poulet\}, \{estOvipare\}) \leq_L (\{Poulet\}, \{aDesPlumes, estOvipare\})$.
La relation d'ordre $\leq_L$ précédemment décrite permet de définir les notions de superconcepts et sous-concepts.

Super-concept et sous-concept

Les ***super-concepts*** d'un concept c_j sont les concepts c_i tels que $c_j <_L c_i$. Comme exemple, nous avons ({Antilope, Sanglier, Poulet, PoissonChat},$\varnothing$) qui est le super-concept de l'ensemble des concepts.
Les ***Sous-concepts*** d'un concept c_j sont les concepts c_i tels que $c_i <_L c_j$. Pour illustrer cette notion, nous avons le concept ({Antilope, Sanglier},{aDesPoils, aDesDents}) qui est un sous-concept du concept ({Antilope, Sanglier, PoissonChat},{aDesDents})

Extension et intension simplifiées

L'***extension simplifiée*** d'un concept (X, Y) est l'ensemble $X' \subseteq X$ des objets (respectivement des attributs) du concept qui n'apparaissent pas dans l'extension de ses sous-concepts.

L'***intension simplifiée*** d'un concept (X, Y) est l'ensemble $Y' \subseteq Y$ des attributs du concept qui n'apparaissent pas dans l'intension de ses super-concepts.

On nommera ***treillis simplifié*** les représentations de treillis comportant l'intension et l'extension simplifiées des concepts. Les treillis simplifiés peuvent toutefois devenir moins pratiques quand le nombre de concepts est élevé, car il faudra naviguer dans le treillis pour connaître tous les éléments situés dans l'extension et l'intension d'un concept donné.

Plusieurs algorithmes de génération de concepts et de construction de treillis ont été proposé. Comme nous le verons dans la suite, certains algorithmes se limitent à l'obtention des concepts sans construire le treillis.

3.2 Algorithmes de construction de treillis

Dans cette section, nous présenterons quelques méthodes de construction de treillis de Galois. Nous allons décrire les quatre algorithmes les plus répandus de la construction de treillis qui sont les rectangles maximaux de la relation binaire.
La table 3.4 nous donne une liste d'algorithmes classifiés [KO01].

	m1	**m2**	**m3**	**m4**	**m5**	**m6**	**m7**	**m8**	**m9**	**m10**
Bordat						x	x			
NextClosure(Ganter)		x					x			
Close by One		x							x	
Lindig					x				x	
Chein		x	x					x		
Nourine	x				x				x	
Norris	x	x							x	
Godin	x		x	x					x	
Dowling	x	x							x	
Titanic										x

TABLE 3.4 – Propriétés des algorithmes de construction des treillis de Galois : m1-incémental, m2-basé sur l'ordre lexical, m3-divise l'ensemble des concepts en plusieurs parties, m4-utilise la fonction de hachage, m5-utilise une structure d'arbre auxiliaire, m6-utilise un attribut cache, m7-calcule les intensions comme une séquence d'intersection, m8-calcule les intersections des intensions déjà générés, m9-calcule l'intersection des intensions des objets, m10-utilise des supports d'ensemble d'attributs

Parmi ces algorithmes, seulement quelques uns construisent au fur et à mesure le diagramme de Hasse. Nous pouvons citer les algorithmes de : Bordat, Godin, Lindig et Nourine. Mais comme nous le verrons plus tard, il existe des méthodes permettant de construire le diagramme à partir d'une liste de concepts.
Dans la suite, nous allons détailler quelques uns de ces algorithmes [GDP+90a] . Il existe deux grands groupes d'algorithmes : les algorithmes incrémentaux et les algorithmes non incrémentaux [ZEN04] .

3.2.1 Algorithmes incrémentaux

Les algorithmes incrémentaux construisent le treillis au fur et à mesure qu'un nouvel objet arrive. Ils comprennent donc une phase d'initialisation, suivie, à chaque nouvel objet entrant, d'une phase de mise à jour.

Algorithme de Godin

L'algorithme de Godin est basé sur la notion de *noeud générateur*. Un nœud (A_i, B_i) est dit générateur pour le nouveau n*o*eud (A_j, B_j) si et seulement si (A_i, B_i) est une borne supérieure de $\{\{ (A_k, B_k) \in L$ tel que $B_j = B_k \cap o\prime \}\}$. L étant le treillis courant, o le nouvel objet et $o\prime$ ses attributs.

Algorithme de Norris

Dans son algorithme, Norris considère le tableau de contexte sous sa représentation binaire. Il parcours la table ligne par ligne et construit l'ensemble des rectangles maximaux L_k après examen de k lignes. Dans cette version , initialement L ne contient qu'un seul rectangle $x_1 \text{x} f(x_1)$. Après examen de toutes les lignes, on obtient l'ensembles des concepts. L'algorithme de Norris, tout comme celui de Godin est incrémental, il incorpore les nouvelles instances une par une.

Algorithm 1 Algorithme de Norris

Require: $X_i \subseteq E$ set of objects, $Y_i \subseteq F$ set of attributes

1: $L \leftarrow x_1 \text{x} f(x_1)$
2: **for** $k \leftarrow 2$ to n **do**
3: **for** $X_i \text{x} Y_i \in L$ **do**
4: **if** $Y_i \subset f(x_k)$ **then**
5: X_i x $Y_i \leftarrow (X_i \cup \{x_k\})$ x Y_i
6: mark x_k
7: **else**
8: **if** $(X_i \cup \{x_k\})$ x $(Y_i \cap f(x_k))$ is maximum **then**
9: $L \leftarrow L + (X_i \cup \{x_k\})$ x $(Y_i \cap f(x_k))$
10: **end if**
11: **end if**
12: **end for**
13: **if** x_k is not marked **then**
14: $L \leftarrow L + x_k$ x $f(x_k)$
15: **end if**
16: **end for**
17: **return** L

3.2.2 Algorithmes non-incrémentaux

Nous allons présenter les deux algorithmes que nous avons implémentés dans le cadre de notre travail.

Méthode de Chein

L'algorithme de Chein est un algorithme non incrémental mais itératif fondé sur la proposition suivante :
Proposition : Soit X_1xY_1 et X_2xY_2 deux éléments de L. Le rectangle $(X_1 \cup X_2)$x$(Y_1 \cap Y_2)$ est un élément de L si et seulement s'il est maximal.
L'ensemble initial, est l'ensemble des rectangles à une ligne correspondant à chaque élément x_i de O, les colonnes étant les éléments de A qui lui sont liés ($f(x_i)$). A chaque étape k, partant d'un ensemble L_k, on construit l'ensemble L_{k+1} par combinaison de deux éléments de L_k. Les éléments de L_k inclus dans au moins un élément de L_{k+1} ne sont pas maximaux, ils sont supprimés. L'algorithme s'arrête quand L_{k+1} a au maximum un élément. A chaque étape k, les éléments non supprimés de L_k constituent les rectangles maximaux.

Algorithm 2 Algorithme de Chein

Require: $X_i \subseteq E$ set of objects, $Y_i \subseteq F$ set of attributes
1: $L^1 \leftarrow x_i$x$f(x_i)_{i=1,..n}$
2: $k \leftarrow 1$
3: **while** $|L^k| > 1$ **do**
4: $L^{k+1} \leftarrow \emptyset$
5: **for** all $i < j$ indices of non marqued elements of L_k **do**
6: $Y_{ij} \leftarrow Y_i \cap Y_j$
7: **if** $Y_{ij} \neq \emptyset$ **then**
8: **if** $Y_{ij} \in L^{k+1}$ **then**
9: $X_{ij} \leftarrow X_{ij} \cup X_j$
10: **else**
11: $L^{k+1} \leftarrow L^{k+1} + X_i \cup X_j$x Y_{ij}
12: **end if**
13: **if** $Y_{ij} = Y_i$ **then**
14: mark X_ixY_i of L^k
15: **end if**
16: **if** $Y_{ij} = Y_j$ **then**
17: mark X_jxY_j of L^k
18: **end if**
19: **end if**
20: **end for**
21: **end while**
22: **return** L^{k+1}

Méthode de Bordat

Cet algorithme non incrémental est le seul qui construit simultanément les éléments du treillis et les arêtes du diagramme de Hasse. L'algorithme consiste à construire la liste des rectangles maximaux [2] en partant du supremum du treillis. Il a une approche descendante, il s'appuye sur la relation de couverture de la relation d'ordre $\leq$. Pour chaque rectangle, on cherche s'il n'est déjà contenu dans la liste. Sinon on prolonge la liste et dans tous les cas on ajoute un arc au graphe de Hasse. L'algorithme de Bordat utilise une structure d'arbre pour le stockage et la recherche de concepts. L'inconvénient majeur de cette méthode est qu'un concept est engendré autant de fois qu'il a de super-concept.

2. Y_i est une *partie maximale* de $g(Y)$x$(X \backslash Y)$ si et seulement si il n'existe pas de partie Z telle que : $Y_i \subset Z$ et $g(Y_i) \cap g(Y) = g(Z) \cap g(Y)$ [GDP+90b].

Algorithm 3 Algorithme de Bordat

Require: $X_i \subseteq E$ set of objects, $Y_i \subseteq F$ set of attributes

$L \leftarrow$ supremum of the lattice
for all X x $Y \in L$ **do**
  Let $y_1, ..y_p$ elements of $F \backslash Y$ indexed in the growing order of the cardinal of $g_X(y_i)$ where $g_X(y_i) \leftarrow g(y_i) \cup X$
  for $i < j$ such as i in 1 to p **do**
    if $g_X(y_i) \subset g_X(y_j)$ **then**
      $c(i) \leftarrow 0$
    else
      if $g_X(y_i) = g_X(y_j)$ **then**
        $c(i) \leftarrow j$
      else
        $c(i) \leftarrow -1$
      end if
    end if
  end for
  for $i \leftarrow 1$ to p **do**
    $Y' \leftarrow \{y_i\}$
    $j \leftarrow i$
    while $c(j) > 0$ **do**
      $Y' \leftarrow Y' \cup \{y_j\}$
      $k \leftarrow j$
      $j \leftarrow c(j)$
      $c(k) \leftarrow 0$
    end while
    if $c(j) = -1$ and $g(Y')$ x $(Y \cup Y') \notin L$ **then**
      $L \leftarrow L + g(Y')$ x $(Y \cup Y')$
    end if
  end for
end for
return L

Chapitre 4

Projet ARSA : Un réseau social pour l'administration

4.1 Le projet

Les réseaux sociaux sont devenus un outil important dans notre société. Ils sont à nos jour l'un des principaux moyens de communication. Mais sont encore peu présent dans les services administratifs.
En 2009, suite à un appel à projets concernant le développement de services Web innovants dans les systèmes d'information des entreprises et des administrations lancé par madame Nathalie Kosciusko-Moziret alors sécretaire d'Etat chargée de la Prospective et du Développement de l'économie numérique en France, quarante quatre dossiers ont été retenus. Parmi ceux-ci, figure celui de l' *Analyse des Réseaux Sociaux de l'Administration* (ARSA). Ce projet, d'une durée initiale de deux ans avait pour but de créer un outil d'analyse des interactions de l'administration, de développer des applications de réseaux sociaux adaptées aux besoins des administrations et des collectivités locales. La ville d'Antibes a été choisie comme ville de lancement de ce projet. Cette ville a été choisie parce qu'elle jouit de liens particuliers avec le centre de recherche SAP. De plus, la taille de la ville [1] correspond à la taille moyenne d'une ville française.
Ce projet se structure en quatre volets [2] :

- Une adaptation d'outils d'analyse de réseaux sociaux existant pour les administrations.
- Une migration d'outils existant vers une plateforme de cloud-computing,

1. 80 000 habitants en janvier 2010, www.linformaticien.com/actualites/id/7599/une-premiere-application-d-arsa.aspx
2. www.old.pole-scs.org/scs-project51726.fr.htm, dernière viste le 2 juin 2011

ceci afin de faciliter la gestion, la maintenance, la mise à l'échelle et la diffusion de notre outil.
- Une amélioration des algorithmes d'analyses et de requêtage afin d'identifier et de caractériser plus finement les réseaux sociaux.
- Une intégration organisationnelle de l'outil d'analyse, une caractérisation des usages de l'outil, et une méthode d'évaluation de sa performance.

Il est mené conjointement par *SAP Labs*, l'*Ecole Centrale Paris* et le datacenter *Euclyde*.

4.2 Les partenaires

4.2.1 Euclyde

Euclyde est un data center de niveau 4 [3] . Cela signifie qu'il a un taux de disponibilité de 99.995% et qu'il n'a pas besoin d'être arrêté pour des raison de maintenance ; il subit au maximum 25 minutes de coupure par an. Le data center *Euclyde* est situé à Antibes/Sophia et spécialisé dans l'hébergement de systèmes et d'applications. En tant que partenaire du projet, il a pour objectif de concevoir, mettre en place et héberger les plates-formes nécessaires par le biais de plusieurs serveurs virtuels répartis en Cloud. Magdi Houry, Directeur d'Euclyde expliquera que "Afin de garantir le contrôle du budget, il était nécessaire d'établir une architecture modulable qui permettrait à la plate forme d'évoluer automatiquement au fur et à mesure de la croissance du nombre de requêtes".

4.2.2 SAP Labs

ARSA est un projet éssentiellement porté par *SAP Labs*, car il se base sur l'application *Social Network Analyser* (SNA) issu du centre d'innovation SAP BusinessObjects qu'il doit faire évoluer. SNA combine à la fois des technologies d'analyse de données et de visualisation. Initialement, SNA a pour but de faciliter la recherche des compétences internes et la création

3. Le standard TIA 942 définit classe les datacenter en 4 niveaux suivant leurs disponibilités :
- Niveau 1 : 99.671% de disponibilité et 29h de coupure par an ;
- Niveau 2 : 99.741% de disponibilité et 23h de coupure par an ;
- Niveau 3 : 99.982% de disponibilité et 1h45 de coupure par an ;
- Niveau 1 : 99.995% de disponibilité et 25min de coupure par an.

des meilleures équipes possibles en fonction des projets à réaliser. SNA permet de réaliser une cartographie d'un réseau social et d'analyser les liens entre les personnes sous différents aspects. A qui reporte telle personne dans l'entreprise ? Dans quels projets l'interlocuteur travaille ? Un utilisateur peut ainsi appréhender les différents engagements et liens dans l'entreprise. C'est grâce à ses capacités en datamining que l'application décrypte des relations explicites ou implicites, permettant, à partir du traitement des informations provenant des systèmes corporate ou sur le web, de visualiser les relations qui existent entre salariés, les relations hiérarchiques, l'appartenance à un département... La solution propose également ce que certains experts SAP appellent "la recherche par facettes", ce terme désigne une recherche multicritère afin de mieux cibler les profils qui répondront le mieux au défi métier à relever (au regard de la fonction occupée, des langues parlées ou écrites, du département, de la spécialité métier ...). Une fois les profils découverts, une console affiche la carte d'identité du salarié (nom, e-mail,...), ainsi que son réseau de relations (managers, collaborateurs, clients,...). Ce concept a intéressé l'administration française, ce qui les a conduit à retenir le projet ARSA soumis par SAP dans le cadre du plan numérique 2012. Le projet ARSA est donc basé sur une adaptation de l'application SNA aux besoins de l'administration. Le projet ARSA sera mené avec la ville d'Antibes, ville dans laquelle le lancement de la phase pilote avait été prévu pour le 18 février 2011. C'est dans ce sens que Gilles Logeais, directeur de recherche de SAP labs dira que "l'administration sert de poisson pilote pour adapter le logiciel".

4.2.3 Ecole Centrale Paris

L'Ecole Centrale Paris aura pour rôle de faire avancer les algorithmes associés au projet. Elle s'occupe de la partie R & D. Dans le cadre de ce projet, le laboratoire MAS (Mathématiques Appliquées aux Systèmes), précisément l'équipe de Business Intelligence avait pour but de développer les algorithmes nécessaires au projet. Ces algorithmes devraient évoluer à l'avenir selon les demandes des administrations.

4.3 Les applications

Nous pouvons voir plusieurs applications du projet ARSA. Si nous considérons la ville pilote, à savoir la ville d'Antibes, comme besoins exprimés, nous notons entre autres : l'amélioration de la recherche par les utilisateurs internes des compétences présentes dans les services de la collectivités, l'amélioration de la visibilité à l'extérieur de l'organisation, l'analyse des communautés et

des différentes tendances dans les interactions avec la la ville pour déterminer l'importance des services à fournir.
Dans la pratique, nous avons plusieurs scénarios. L'un concerne les relations qu'a la ville avec ses fournisseurs. Un second est l'analyse des relations avec les associations. Le service des sports de la ville a pu par exemple affirmer que la voile était le premier sport pratique dans la ville au détriment de certains sports qui sont plus médiatiques.
SAP souligne que ARSA pourra être utilisé dans le secteur public dans le domaine de la gestion des appels d'offres publics qui font intervenir un grand nombre de fournisseurs. Un autre domaine est celui des relations entre les institutions et les citoyens, notamment pour l'amélioration de la transparence dans l'administration. Dans ce sens, SAP envisage de mettre en œuvre des dispositifs plus standard, tels que la visualisation d'organigrammes [4].

Comme nous l'avions souligné plus haut, le but du projet ARSA était de proposer un outil d'analyse des interactions de l'administration. Moyennant quelques modifications, l'application SNA de SAP proposait un début de solution à l'analyse des interaction des différents échanges internes à l'administration, mais aucun outil n'était encore dédié à l'analyse et à la réputation que l'administration avait du monde extérieur. Pour répondre à cette préoccupation, l'Ecole Centrale Paris a proposé l'application *Evarist*.

4. www.linformaticien.com/actualites/id/7599/une-premiere-application-d-arsa.aspx, dernièreviste le 2 juin 2011

Chapitre 5

EVARIST

L'analyse d'opinion sur le web est un domaine de recherche en pleine croissance. En effet, plusieurs plateformes sur le web mettent à la disposition des utilisateurs des espaces leur permettant de s'exprimer, de laisser leur opinion ou de consulter les avis des autres utilisateurs sur un sujet donné. C'est le cas des sites de critiques littéraires [1], de films [2] ou de shopping. De même, de plus en plus de personnes et d'entreprise se soucient de l'image, de la réputation qu'ils ont sur le web. Cet intérêt de plus en plus croissant de l'analyse d'opinion sur le web est motivé par l'apparition et le développement des blogs et des réseaux sociaux qui sont les outils les plus utilisés pour l'expression d'opinions sur Internet. Les plus récents sont les sites de microblogage. Parmi ces réseaux sociaux et sites de microblogage, Facebook et Twitter sortent du lot. Ils sont les plus utilisés, et les différentes fonctions de partage et de transfert d'information dont ils disposent permettent d'observer en "temps réel" la formation d'opinion, et ainsi de se faire une idée des tendances futures [PLG11].
Ce chapitre s'organise de la façon suivante : dans la section 1, nous allons présenter quelques outils de buzz monitoring proposés. Dans la section 2, nous présenterons un prototype d'outil d'e-reputation basé sur les treillis de Galois nommé EVARIST.

5.1 E-buzz Monitoring et E-reputation

Avec l'essor des communautés, l'on se rend compte qu'il devient indispensable de bien gérer sa réputation sur le web. En effet, nous avons rencontré

1. Exemple : www.critiqueslibres.com
2. Exemple : www.critikat.com

certains cas où un employé se fait licencier sur base de publication [3] sur son profil. La question qui en découle est de savoir comment gérer cette e-réputation et surveiller celle des autres. Rapellons que selon Nicolas Bariteu, l'e-réputation ou identité numérique est " l'image réelle ou fantasmée que les internautes se font d'une institution, d'une entreprise, d'une marque ou d'une personne sur tous les types de supports numériques (médias, réseaux sociaux, forums, messagerie instantanée...)" [INF]. Il existe plusieurs outils de buzz monitoring. Dans cette section, nous allons présenter quelques outils d'e-reputation[ROL09].

1. **Samepoint**
   Samepoint est un méta-moteur de recherche de conversations sur internet(blogs, réseaux sociaux et contenus générés par des utilisateurs). Ces conversations ne sont pas toujours indexées par les moteurs de recherche classiques. Pour un sujet donné, il propose de consulter en temps réel les commentaires et les messages postés sur Twitter, sur Facebook et même sur LinkedIn. Samepoint met à la disposition des utilisateurs une saisie d'écran, les liens sources, la traduction et surtout le "Social Tone" qui permet une analyse succinte des mots positifs et négatifs dans un message [4].

2. **Trackur**
   Trackur [5] est un outil permettant d'analyser le web afin de rechercher le nom d'une entreprise, d'un produit, d'une ville, d'un pays ou d'une actualité. L'utilisation de ce service se fait via un abonnement. En fonction du montant de l'abonnement, l'utilisateur a le droit de surveiller un nombre de mots clés. Par exemple pour un coût de 88 dollars, il est permit de surveiller 5 mots clés toutes les 6 heures. La formule de base coûte 18 dollars et permet la surveillance d'un mot toutes les 12 heures. Trackur vous informe de l'actualité liée aux mots que vous suivez par email [6].

3. **Google alerte**
   Lancé en 2004, ce service permet d'avertir par email quand des informations publiées en ligne correspondent à un sujet indiqué par l'utlisateur. Il permet de suivre l'évolution de l'actualité sur un sujet donné ou sur une personnalité. Comme Trackur, Google envoie les résultats de la veille par email à l'utilisateur. Ce dernier a la possibilité de choisir la

3. Par exemple en critiquant son employeur ou en dévoilant des informations sensibles.
4. www.samepoint.com
5. www.trackur.com
6. www.actulligence.com/2008/07/14/trackurcom-suveillance-de-reputation-en-ligne/ ; dernière visite le 21 juin 2011

fréquence de réception des résultats. Cependant, pour gagner en efficacité et accroître la pertinence des résultats fournis par cet outil, il est important de connaître Google et sa syntaxe de recherche. Une limite de ce service réside dans le délai de disponibilité de l'information. En effet, si l'utilisateur ne consulte pas immédiatement ses mails liés à l'alerte, ceux-ci seront archivés et l'accès à l'information pourra devenir payant [7].

4. **Sentiment Metrics**
   Sentiment Metrics propose de scruter le web pour analyser votre réputation en ligne. Ses sources sont diverses : blogs, forums, sites d'actualité, communiqués de presse. Cet outil n'offre pas de flux RSS, mais comme beaucoup d'autres, il propose des alertes mail. Sa particularité est qu'il met à la disposition des utilisateurs des tableaux et des graphiques (courbes de popularité) pour meilleure compréhension de la réputation [8].

Cette liste est loin d'être exhaustive, il existe une multitude d'outils d'e-réputation que nous ne pouvons tous citer ici. La principale limite que la majorité de ces outils rencontrent est la gestion de la langue. De plus, très peu offrent l'analyse des phrases dans lesquelles apparaissent les mots cherchés, proposant ainsi de l'information brute sans analyse. Dans la suite, nous allons présenter un outil qui utilise l'analyse formelle et les treillis de Galois pour analyser l'e-réputation sur Twitter : EVARIST (E-buzz VAluable Radar for Internet and Social neTworks).

5.2 EVARIST

L'outil proposé a pour but d'analyser un groupe de tweets afin d'y repérer les termes et groupes de termes les plus tweetés. Cette application est développée sous l'environnement R. Nous présenterons de manière détaillée l'environnement de développement dans le chapitre suivant. Les auteurs d'Evarist définissent l'analyse du groupe de tweets en quatre étapes. Afin de mieux présenter ces différentes étapes, nous allons appliquer la démarche en faisant une recherche sur le mot clé "♯ecoli".

7. www.les-infostrateges.com/article/0606263/les-alertes-google-un-outil-au-service-de-la-veille
8. www.sentimentmetrics.com

5.2.1 Récupération des tweets

La récupération des tweets associés au mot cherché se fait principalement avec *searchTwitter* disponible dans le package *twitteR*[9]. TwitteR est un package de l'environnement R destiné à la gestion des informations sur Twitter[10].
La récupération des tweets se fait par l'appel de la fonction searchTwitter. Elle est définie comme suit

searchTwitter(searchstring, n, lang, since, until, locale, geocode, sinceID,...)

où :

- *searchstring* : représente le mot ou le groupe de mots recherché sur le réseau Twitter.
- *n* : correspond au nombre de tweets que l'on veut récupérer.
- *lang* : si on affecte une donnée différente de NULL à ce paramètre, alors les tweets récupérés seront ceux rédigés dans la langue spécifiée.
- *since* : si sa valeur n'est pas NULL, alors les tweets sont restreints à ceux émis depuis la date spécifiée.
- *until* : permet de restreindre la recherche des tweets à ceux émis jusqu'à une certaine date.
- *locale* : définit les paramètres régionnaux de la recherche des tweets
- *geocode* : retourne les tweets des utilisateurs situés dans un rayon donné par la longitude et la latitude.
- *sinceID* : retourne les tweets ayant un ID supérieur à celui spécifié.
- ... : permet de définir les arguments optionnels.

Dans l'exemple proposé, le souhait est de récupérer 50 tweets associés au mot "ecoli". La langue, la date et la localisation n'ont pas d'importance. La fonction s'écrit donc :

searchTwitter("♯ereputation",n=50)

Ci-dessous, nous présentons les 10 premiers tweets obtenus[11] :

1. gmopundit : Biofortified http ://bit.ly/iC7PZz Flawed ♯foodsafety standards led to ♯Ecoli deaths ?
2. nehayanashed : RT @i_ naguib : Great. Coz we don't have enough problems already. Hopefully untrue. http ://t.co/GQEjQyd ; ♯EHEC ♯ecoli

9. www.cran.r-project.org/web/packages/twitteR/index.html
10. Microblog
11. Requête exécutée le 28 juin 2011 à 9h09

3. gmopundit : ♯GMO Pundit a.k.a. David Tribe http ://bit.ly/k2lw3A Flawed ♯foodsafety standards in Germany caused ♯Ecoli deaths ?
4. worldfoodlinks : United States : A Decade of Inaction at ♯USDA on Non-O157 E. coli http ://goo.gl/fb/NY8v5 ♯ecoli ♯foodsafety
5. FluTrackers : FluTrackers cautions about blaming ♯EHEC ♯ecoli European outbreak on seeds imported from ♯Egypt.We have seen no similar outbreaks there. ♯fb
6. Crof : ♯EHEC patients have a year of follow-up http ://t.co/rqxgZ5c ♯ecoli"
7. TrusonOrganics : RT @PamelaDrew : About time we saw ♯Wired with a food industry piece that didn't favor ♯AgriBiz http ://t.co/8OxB3GG Tx @geeknik ♯Ecoli ♯cafo ♯H1N1 ♯flushot"
8. "AdventureDoc : @TravelSafety : TravelSafety Grilling kills small amounts of ♯Ecoli http ://t.co/pyekaJD ♯tvlmed good to know !
9. travellersalert : TravelSafety Grilling kills small amounts of ♯Ecoli http ://-bit.ly/kRDHF2
10. TravelSafety : TravelSafety Grilling kills small amounts of ♯Ecoli http ://-bit.ly/kRDHF2

Comme nous le remarquons, les tweets obtenus contiennent plusieurs termes qui ne sont pas utiles pour la compréhension et l'analyse d'un tweet. Il est donc nécessaire de les supprimer.

5.2.2 Nettoyage des tweets

Le nettoyage des tweets consiste en la suppression des mots de liaison, des ponctuations, des opérateurs arithmétiques, des chiffres, des caractères et signes spéciaux.
Comme ensemble des signes à supprimer nous avons :
{","," ;"," :"," ?"," ."," /","+","=","~","*","\\","RT","RT :","<",">","_ ","$","& ","|",". . .","@","","^","¨","(",")"," !", "{","}","[","]","%","0","1","2","3","4","5","6","7","8","9"}.
En ce qui concerne le traitement des mots, les auteurs d'Evarist considèrent principalement deux langues : les français et l'anglais. Ils définissent ainsi deux listes de stemmers à supprimer. Un stemmer étant un algorithme qui supprime les suffixes dérivationnels pour réduire les différentes formes d'un mot à leur racine[12]. Par exemple, en anglais les mots *"formula", "formulate", "formulation"* et *"reformulate"* seront remplacés par le mot racine

12. www.alx2002.free.fr/utilitarism/stemmer/stemmer_ fr.html♯use

"formula".
En résumé, l'ensemble des termes à supprimer dans les tweets traités seront les caractères et signes spéciaux, les stemmers en anglais et en français. Le résultat obtenu par ce nettoyage nous permet de construire la table de contexte associée à cet ensemble de tweets.

5.2.3 Table de contexte

Dans la table de contextes qui dérive de l'ensemble des tweets nettoyés, les objets sont les tweets et, les attributs sont les mots contenus dans les tweets nettoyés. En effet, après avoir nettoyé les tweets, on y extrait les mots résultats et on supprime les doublons.
La matrice obtenue sur base de notre exemple est caractérisée par 50 lignes représentant le nombre de tweets et 76 colonnes représentant les différents mots extraits des tweets.

	biofortified	flawed	♯foodsafety	standards	led	♯ecoli	deaths	problems
1	1	1	1	1	1	1	1	0
2	0	0	0	0	0	1	0	1
3	0	1	1	1	0	1	1	0
4	0	0	1	0	0	1	0	0
5	0	0	0	0	0	1	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	1	0	0
8	0	0	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0
10	0	0	0	0	0	1	0	0

TABLE 5.1 – Sous table de contexte Ecoli

Le tableau 5.1 illustre une partie de la matrice de contexte obtenue. Il présente les 8 premières colonnes de la table de contexte.

5.2.4 Calcul du treillis de Galois

Une fois la matrice de contexte construite, l'étape suivante est la construction du treillis de Galois. L'application utilise un package dédié à cette tâche. Il s'agit du package *galois* que nous présenterons de manière plus détaillé dans la suite de notre exposé. Ce package implémente plusieurs algorithmes de construction de treillis.

5.2.5 Visualisation des résultats

La figure 5.1 présente le treillis associés aux tweets attachés au mot *"ecoli"*. La surface d'un n*oe*ud est proportionnelle au nombre de tweets qu'il contient. Nous constatons qu'il est difficile d'afficher simultanément tous les concepts et leurs attributs de façon lisible. Pour résoudre ce problème de lisibilité, les auteurs proposent de ne sélectionner que les concepts dont la taille dépasse un certain seuil. Ce qui est logique avec la notion de buzz qui correspond au groupe de mots les plus tweetés.
Malgré l'application d'un seuil, nous constatons que la lisibilité n'est toujours

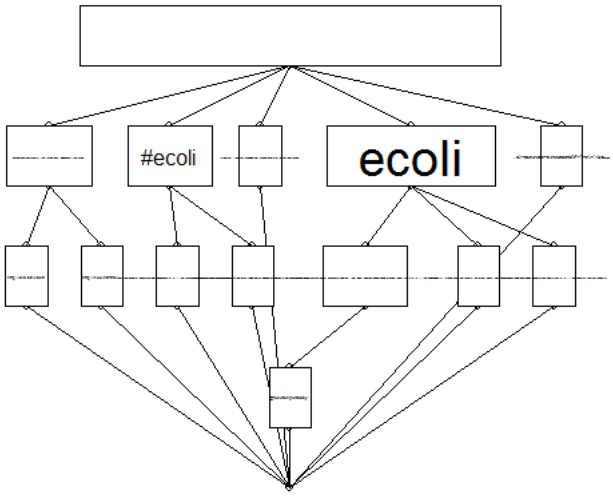


FIGURE 5.1 – Treillis associé aux tweets contenant "ecoli"

pas très bonne. Ceci est dû au fait que nous disposons d'attributs formés de mots assez longs.
Une autre forme de visualisation est l'utilisation des nuages de tags. Dans ce

type de visualisation, la taille d'un attribut varie en fonction de son importance dans le réseau. Cependant, la limite de cette réprésentation vient du fait qu'il n'y a plus de lien entre les différents concepts et leurs sous-concepts. Comme solution à cette limite, certains chercheurs proposent d'afficher les tags associés les uns après les autres.
La visualisation proposée dans Evarist est la fusion des méthodes ci-dessus. Ainsi, les concepts les plus importants sont affichés sous forme de nuage de tags, et les liaisons entre les différents concepts sont représentées par des arêtes. Le sens d'une arrête est du concept vers son sous-concept. En ce qui concerne la disposition des nœuds dans l'espace d'affichage, c'est la méthode de Fruchterman et Reingold [FR91] qui a été implémentée. Elle permet une occupation optimale de l'espace, ce qui améliore la lisibilité des graphes. Pour accroître la lisibilité, les auteurs d'Evarist ont ajouté une allégorie topographique. A cet effet, à chaque concept est associé un niveau. Un mélange de gaussiennes à deux dimensions transformées est utilisé pour la mise au point des niveaux. Les centres sont les coordonnées des centres des tags, et les écart-types sont caractérisés par les hauteurs et les largeurs des tags. Enfin, pour que la hauteur soit proportionnelle à la taille des concepts, il a fallu normaliser les hauteurs des gaussiennes en les multipliant par les écart-types et par les hauteurs souhaitées. La fonction topigraphique obtenue est la fonction T telle que :

$$T(x,y) = \sum_{i=1}^{k} \frac{s_i}{2\pi} e^{-\frac{(x-x_i)^2(y-y_i)^2}{2l_i^2 h_i^2}}$$

où :

- k : représente le nombre de concepts affichés,
- x_i et y_i sont les coordonnées du i^e concept,
- l_i et h_i représentent respectivement la largeur et la hauteur du i^e concept,
- s_i est la taille du i^e concept

Le résultat est présenté dans une page HTML reprenant l'ensemble des tweets, le diagramme de Hasse et la liste des concepts construits.

Avec l'essor des réseaux sociaux et des plateformes de micro-blogging, la vitesse d'accès et de diffusion de l'information ne cesse d'augmenter. Twitter se démarque des autres. En effet, depuis son lancement en 2006, il ne cesse de gagner en popularité. Que ce soit pour l'élection de Barack Obama, les émeutes en Egypte ou le séisme au Japon, l'impact de ce service a été indéniable. Les entreprise ne sont pas restés insensibles à cette grande ca-

pacité de diffusion d'information. Selon une étude de "The Global Social Media Check-up", 71% des entreprises européennes figurant dans le classement mondial Fortune 100 ont un compte Twitter. Et parmi celles-ci, 65% tweetent activement [13]. Dès lors, pour les entreprises et les individus voulant gérer leur réputation en ligne, plusieurs outils de monitoring de buzz et d'e-réputation ont été conçus. Dans ce chapitre, nous avons présenté une technique de monitoring qui se base sur les treillis de Galois, et la visualisation des concepts qui y est proposée est une fusion des techniques de nuages de tags et de représentation proportionnelles en réseau topigraphique. Cette technique permet de limiter l'affichage aux concepts les plus importants et permet l'affichage des tags constituants les concepts de manière plus lisible si l'on souhaite afficher directement le treillis. Les liens entre concepts sont représentés par des arêtes dirigées, ceci a pour but de faciliter la lecture du treillis dans le sens où les concepts les plus généraux sont affichés tout en haut, au "sommet". Les arêtes servent alors de piste pour guider l'utilisateur vers des concepts liés. Evarist a été développé sous l'environnement R et utilise le package *galois* implémenté par les concepteurs pour la génération et la visualisation des treillis de Galois.
Dans le chapitre suivant, nous allons présenter de manière détaillée l'environnement de travail nécessaire pour le développement de cet outil.

13. En moyenne 27 tweets par semaine ; http ://pro.01net.com/editorial/519718/informer-diffuser-echanger-les-atouts-business-de-twitter/ ; dernière consulation 01/07/2011 à 8h00

Chapitre 6

Environnement R

Dans cette section, nous abordons le logiciel R qui présente beaucoup d'avantages que ce soit dans le domaine de la recherche ou des entreprises. Dans un premier temps, nous décrirons de manière générale l'environnement R, ensuite nous présenterons l'éditeur de texte que nous lui avons associé lors de nos travaux. Par la suite, nous montrerons comme R gère ses données. Pour finir, nous détaillerons quelques methodes de construction des graphiques.

6.1 Présentation de R

R est à la fois un logiciel et un langage de programmation qui permet de réaliser des analyses statistiques. Il dispose des outils permettant la manipulation des données, des calculs et des représentations graphiques. Il est une sorte de pack muni de[HUI02] :

- un système de manipulation et de stockage de données ;
- des opérateurs de calculs sur les matrices et les tableaux ;
- d'outils d'analyse de données et de méthodes de calcul statistiques ;
- d'outils graphiques pour la visualisation des résultats d'analyse (c'est l'une des raison pour laquelle on l'a choisi) ;
- un langage de programmation permettant d'implémenter les conditions, les boucles, disposant des fonctions d'entrée et de sortie, et proposant

la possibilité de définir des fonctions récursives.

Deux langages ont fortement influencé la mise en oeuvre de R. Il s'agit :

- du langage **S** : développé dans les années 70 par Rick Becker, John Chambers et Allan Wilks pour le compte de AT & T Bell Labs [1] . Le but était de proposer un langage permettant de supporter les activités de recherche dans le département statistique. Il est possible d'acceder à ce langage via le logiciel **S-Plus** [2] ou via l'environnement **R**.
- du langage **Scheme** : langage fonctionnel dont le principe fondamental est la récursivité [3]. Scheme est drivé du langage Lisp[FEE].

R est très semblable à **S**. La majorité des fonctions accessibles sont écrites en **R**. Tous les éléments de R sont présentés sur le site www.R-project.org. Ce site met à la disposition des utilisateurs un ensemble de site fournissant ce qui est nécessaire à la distribution de R : ses extensions, sa documentation, ses fichiers sources et ses fichiers binaires. Cet ensemble se nomme le "Comprehensive R Archive Network" (**CRAN**). Il est similaire à CTAN [4] de LATEX.
R est multi-OS : *Windows, Unix, GNU/Linux, MacOS* et multi-plateforme : 32bits, 64bits, Risc, Sics. De plus, il fait partie du projet GNU [5]

6.2 Les auteurs

R est un environnement Open Source d'analyse statistique et graphique initialement crée par Ross Ihaka et Robert Gentleman [6] [IHA96r].
Depuis mi-1997, une équipe (la "R Core Team") se charge de la maintenance et du développement de R. Aujourd'hui, cette équipe est constituée de 18 personnes.
Etant donné que R est un logiciel libre, cette équipe est soutenue par toute la communauté des développeurs qui proposent de nouvelles fonctionnalités.

1. www.stat.ucl.ac.be/cours/stat2430/documents/manuels_ logiciels/syllabusR.pdf
2. S-Plus est un logiciel commercialisé : www.splus.com
3. Schem : prononcé "skim", et a pour site de reférence : www.schemers.org
4. CTAN : Comprehensive Tex Archive Network.
5. Le projet GNU a pour but de développer un système d'exploitation complet utilisant uniquement des logiciels libres.
6. Ross Ihaka et Robert Gentleman étaient des chercheurs du département des statistiques de l'université d'Auckland de Nouvelle Zélande

6.3 Points forts de R

Le premier point fort que nous pouvons soulever est que le logiciel R propose des fonctions permettant une gestion efficace des tableaux et des matrices. En effet, les opérations de manipulation des tableaux et des matrices ont des temps d'exécutions assez faibles. Le second point fort vient du fait R est un logiciel libre. De ce fait, il bénéficie d'un réseau international de développement, ce qui laisse présager une perpetuelle évolution. De plus, sa facilité de programmation et sa forte utilisation en particulier dans le monde de la recherche sont des atouts. Avec cette forte communauté de participants, R est un logiciel complet[COR09], car presque toutes les méthodes statistiques sont déja implémentées.

6.4 Points faibles de R

R est très sensible à l'utilisation des boucles. Dès qu'on fait appel à un nombre très élevé de boucles, R peut avoir un problème de mémoire. L'utilisation des boucles est très coûteuse en temps de calcul. Il est donc prédérable de limiter l'utilisation des boucles et de privilégier les opérations matricielles. Quant l'utilisation des boucles est inévitable, il est conseillé de les implémenter en C avant de les intégrer dans le code R.
L'environnement R dispose d'une interface d'utilisation. Généralement, à cette interface, on associe un éditeur qui facilite la saisie du code. Il existe plusieurs éditeurs pour le langage R. Nous allons décrire l'environnement R que nous avons utilisé. Mais avant cela, nous allons présenter l'éditeur utilisé pour notre développement : Tinn-R.

6.5 Editeur de code : Tinn-R

Tinn-R est un éditeur de code gratuit et open source. Il a été développé en Delphi 5 et fonctionne uniquement sous Windows. Comme la plupart des éditeurs de code, il propose des colorations syntaxiques pour le code R. En plus, il détecte automatiquement l'existence de l'environnement R sur l'ordinateur. Il dispose aussi des fonctionnalités permettant de soumettre directement le code R à la console. Les sources de l'application sont disponible sur sourceForge.

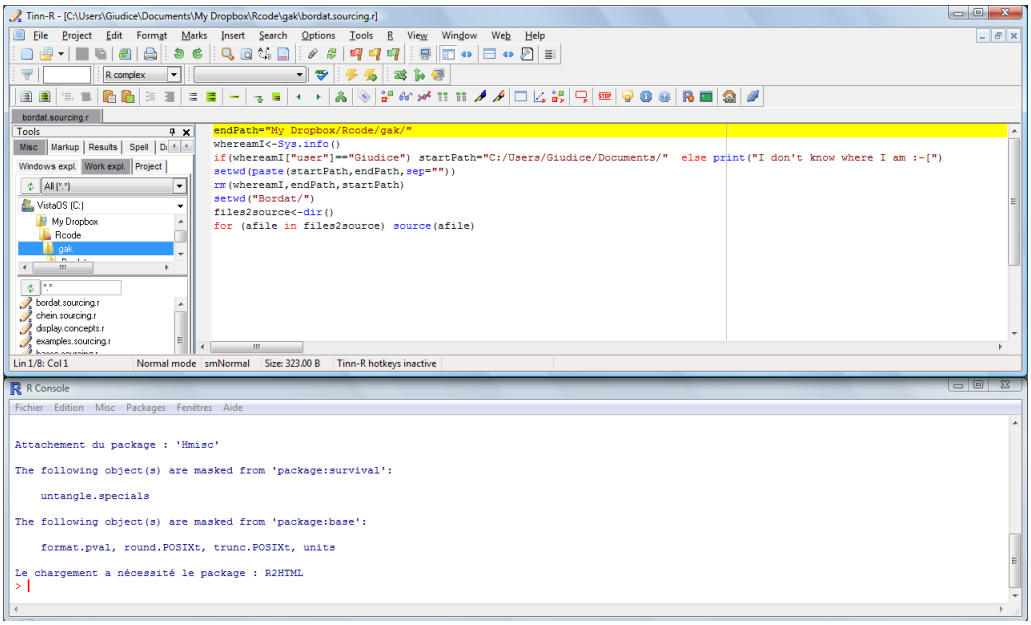


FIGURE 6.1 – Tinn-R

6.6 L'interface d'utilisation de R

Nous présenterons l'interface sous le système d'exploitation Windows, vu que c'est sous ce système que nous avons développer. Sous Windows, le langage R est accessible via l'application **Rgui.exe**. Nous allons décrire la version 2.11.1 de cette application.

Cette application propose une interface simplifiée. Le but étant de fournir aux utilisateurs des raccourcis vers quelques commandes. L'environnement se structure principalement autour d'une barre de menu et d'une console. La barre de menu contient les items suivants :

- **Fichier** : ce menu contient les outils pour la gestion de l'espace de travail. Nous pouvons citer entre autres la sélection du repertoire de travail, le chargement des fichiers sources, la sauvegarde et l'historique de l'environnement de travail.
- **Edition** : ce menu contient les outils nécessaires pour l'édition du code.
- **Misc** : ce menu traite de la gestion des objets en mémoire et permet d'arrêter une procédure en cours de traitement.
- **Packages** : ce menu permet l'installation, la mise à jour, la gestion des librairies via le CRAN.
- **Fenêtre** : ce menu a pour rôle la gestion des différents fenêtres.
- **Aide** : ce menu gère l'accès de l'aide en ligne et aux manuels de

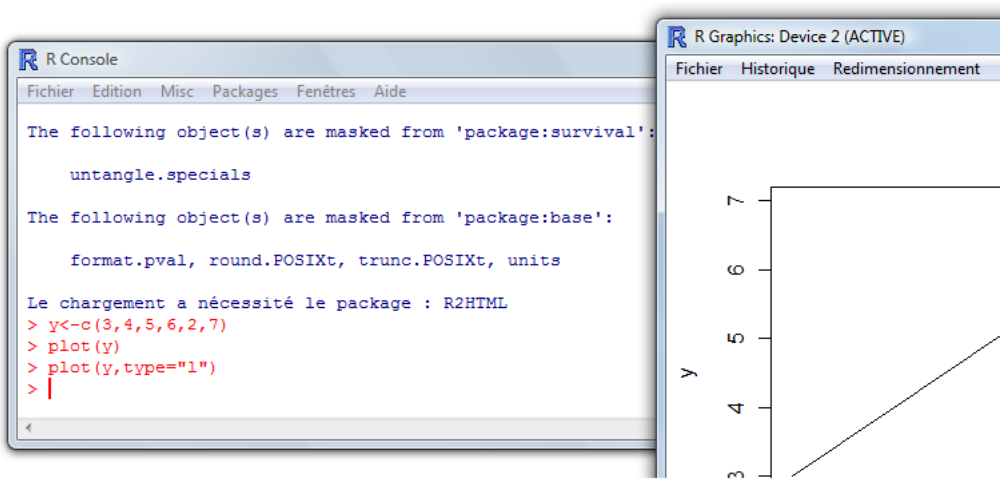


FIGURE 6.2 – Interface d'utilisation de R sous Windows

références de R.

La console quant à elle est la fenêtre où l'on saisie les commandes en entrées et où s'affiche les résultats en mode texte. Suivant la commande exécutée, à cette fenêtre principale peut s'ajouter des fenêtres graphiques et d'informations.

6.7 Les données dans R

R manipule les données sous forme d'objets. Chaque objet est caractérisé par un nom, un contenu décrit par un mode et une structure définie par une classe.
Les différents modes que nous avons dans R sont : *logical, numeric, complex* et *character*.
Quelque soit le mode, les valeurs manquantes sont représentées par **NA**[7].
Comme classe, les principales que nous pouvons citer sont :

- *vector* : pour les vecteurs. Un vecteur étant une collection ordonnée d'éléments de même type.
- *matrix* : pour les matrices. Une matrice ne pouvant contenir que des éléments de même type.
- *array* : pour les tableaux.
- *data.frame* : pour les tableaux de données. C'est une structure analogue à la matrice, à la différence que les colonnes peuvent être de modes différents.

7. NA : not available

- *list* : pour les listes. Une liste peut contenir n'importe quel type d'objets.

6.8 Les graphiques dans R

R permet la gestion des graphiques. Dans cette section, nous présentons la fonctionnalité graphique de base.
La fonction graphique la plus utilisée est la fonction *plot()*. C'est une fonction de haut niveau car elle génère un nouveau graphique dans une nouvelle fenêtre. C'est aussi une fonction *générique* dans le sens où le graphique produit dépend de la classe de son premier argument.
La fonction *plot* a la spécification suivante : *plot(x,y,...)* où :

- x sont les coordonnées des points.
- y sont les coordonnées complémentaires, ils sont optionnels.
- . . .représente la possibilité d'ajouter des arguments à passer à la méthodes. En général ce sont les paramètres graphiques.
  Exemple :
  $z < -ts(matrix(rnorm(300), 100, 3), start = c(1961, 1), frequency = 12)$
  $plot(z)$

La fonction *plot* permet la représentation graphique pour tout type d'objet R. Il existe aussi de nombreuses options disponibles pour le paramétrage et la personnalisations des graphiques. Ces paramètres peuvent se définir directement comme argument de fonctions graphiques ou comme argument de la fonction *par()* que nous ne présenterons pas ici. Il existe d'autres fonctions nécessaires pour la gestion des graphiques sous R, la fonction *help* permettra d'avoir plus de détails.

Dans ce chapitre, nous avons présenté l'environnement R. Cette présentation qui n'est certainement pas exhaustive nous a permis de noter que l'environnement R est muni d'un langage assez intuitif qui peut s'avérer utile dans plusieurs domaines, aussi bien dans la recherche que dans les entreprises. De plus, il dispose d'outils de représentations graphiques et statistiques qui évoluent continuellement à cause de la grande communauté de développeurs qui proposent de nouveaux modules. Pour apporter notre pierre à l'édifice, dans le cadre du projet Evarist, nous avons construit un package R. Ce package est présenté en détail dans le chapitre suivant.

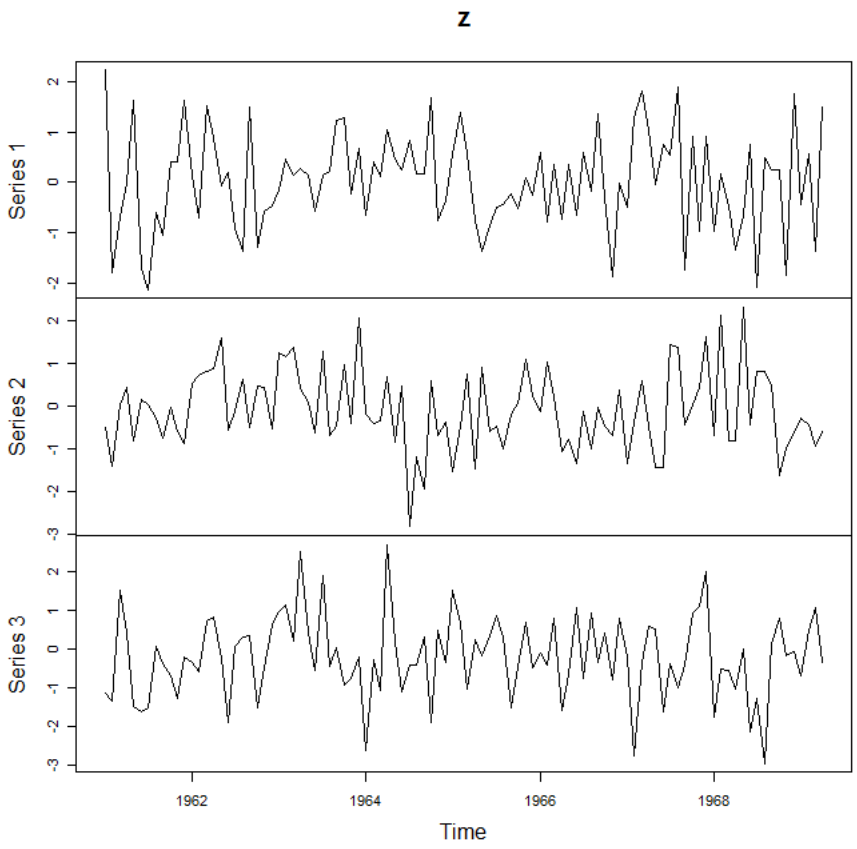


FIGURE 6.3 – Représentation de 3 séries avec la fonction *plot*

Chapitre 7

Package Galois

Comme nous l'avons souligné précédement, l'application Evarist a nécessité l'implémentation d'un package R dans lequel on implémente différents algorithmes de construction de Treillis de Galois.
Un package peut se définir comme étant un fichier compressé contenant un ensemble de fonctions, de documents et de jeux de données nécessaire pour illustrer ces fonctions. Un package a pour but de faciliter l'utilisation et le partage des fonctions. Dans l'environnement R, c'est le CRAN [1] qui est chargé de la gestion et de la validation des packages R. Actuellement on dénombre près de 3129 [2] packages sur le site du CRAN. On retrouve aussi des packages sur les pages webs personnelles des développeurs.
Dans ce chapitre, nous allons présenter de manière détaillée tous les modules du package Galois. Mais avant, nous allons rappeler comment se contruisent les packages sous l'environnement R.

7.1 Objectifs

Nous avons un double objectif : proposer un package R d'analyse et de visualisation du réseau social Twitter.

7.2 Création de package R

Avant de présenter la procédure de création d'un package R, rappelons que le format du package dépend de la plateforme sur laquelle on travaille.

1. Site officiel : www.cran.r-project.org/
2. Dernière visite le 16 juillet 2011

Sous Windows, le package a l'extension ".zip" et se constitue de fichiers précompilés. Alors que sous Linux, l'extension est ".tar.gz" et le package est constitué de codes sources. Quant à nous, nous avons travaillé sous Windows. La création d'un package R se fait en plusieurs étapes [Bai06]. Avant de passer à la création du package, il faut créer dans R les fonctions et objets devant figurer dans ce dernier.

1. Création du répertoire de base : ceci se fait avec la fonction "package.skeleton". Cette fonction permet de créer un modèle de package contenant divers répertoires et squelettes de fichiers d'aide. Les principaux éléments créés sont :
   (a) Le répertoire *R* : ce répertoire contient les fichiers des codes R compilés.
   (b) Le répertoire *data* : il contient tous les fichiers correspondant à un jeu de données. Ces fichiers ont pour extension *.rda*
   (c) Le répertoire *man* : contient les fichiers de documentation. Ces fichiers ont une structure proche de celle des fichiers Latex. Ces fichiers sont sauvegardées au format *.Rd*[3]. Ces fichiers peuvent être compilé en Latex, Pdf, Html, Html Help[4] , ...
   (d) Le répertoire *src* : contient le codes compilés écrit en langage autre que le langage R. Dans ce cas, il contiendra les fonctions C que nous avons codées.
   (e) Le fichier *Description* : fournit une brève description du package. En plus, il donne des renseignements tels que l'auteur, la version et la date de création du package.
   (f) Le fichier *Read Me*
2. Ajout des fichiers pour l'incorporation des codes C, C++ ou même FORTRAN
3. Modification des fichiers d'information sur le package
4. Vérification : cette étape a pour but de vérifier que les fonctions sont bien documentées et que les exemples fournies fonctionnent.
5. Construction du package

7.3 Package Galois

Comme nous l'avons souligné, le package Galois est le principal module nécessaire au projet Evarist. C'est dans ce package que sont développés les

3. Rd : "R documentation format"
4. Format standard d'aide pour les applications windows.

fonctions de génération de Treillis et de construction. Rappelons que l'on fait appel à ce package une fois que les tweets obtenus ont été néttoyés et que la matrice de contexte a été contruite.
Dans la suite, nous allons présenter ce package. Les fonctions de ce package sont appliquées à la matrice contexte que nous nommons ici *table de contexte*. Cette table de contexte a les caractéristiques suivantes : en ligne, sont disposés les objets et en colonne sont disposés les attributs des objets.

7.3.1 Génération de concepts de Treillis

Plusieurs fonctions ont été implémentées pour la génération des concepts de Treillis. Nous allons détailler ces fonctions. Mais avant, nous allons expliquer les principales sous-fonctions nécessaires à la constructions des fonctions de génération de concepts.

Sous-fonctions

Nous présentons dans cette section les principales sous fonctions qui ont été utilisées pour l'implémentation des algorithmes de génération de treillis. Parmi ces fonctions, nous en avons qui ont pour rôle la représentation graphique du diagramme de Hasse. Certaines de ces fonctions correspondent aux opérations définies dans le chapitre consacré aux treillis de Galois.

1. **ext** : étant donné une table de contexte, cette fonction calcule l'extension d'un ensemble d'objets. Elle a pour but de retrouver l'ensemble des objets ayant certaines caractéristiques spécifiées. Cette fonction est l'implémentation de l'opération f définie dans le chapitre prévu à l'analyse formelle de concepts.
2. **int** : appliquée à une table de contexte, cette fonction calcule l'intension d'un ensemble d'attributs. Partant d'un ensemble d'objets, cette fonction permet de déterminer les attributs que ces objets ont en commun. Cette fonction correspond à l'opération g définie dans le chapitre dédié à l'analyse formelle de concepts.
3. **attrib2concept** : étant donné un ensemble d'attributs et une table de contexte, cette fonction construit le concept associé. Rappelons qu'un concept est une structure composée d'un ensemble d'objets et d'un ensemble d'attributs communs aux objets du premier ensemble. Cette fonction implémente l'opération X''. [5]

5. Voir fermeture Chap 3

4. **obj2concept** : partant d'un ensemble d'objets et d'une table de contexte, cette fonction construit un concept. Cette fonction est l'implémentation de l'opération Y'' [6].
5. **build.gl** : est la fonction qui construit la structure galois qui sera utilisée pour l'élaboration du diagramme de Hasse. Elle fournie une structure de donnée constituée de :
   - La liste des concepts générés.
   - Une matrice de deux colonnes représentant les liens entre les *superconcepts* (les parents), ici les éléments de la première colonne et les *sousconcepts* (les fils), les élemnts des la deuxième colonne de la matrice . Dans l'implémentation, cette matrice se nomme *edge.list*.
   - Le contexte sous forme de matrice booléenne utilisé pour la génération des concepts.
   - Un vecteur de niveau qui permet de définir à quel niveau hierarchique se situera chacun des concepts.
   - Un vecteur pour la taille de chaque niveau à représenter.
   - Un vecteur pour la taille des concepts. Ce vecteur sera utilisé pour la représentation des n*oe*uds du diagramme.
   - Les noms des attributs et des objets du contexte.
6. **plot.galois.lattice** : fonction pour le tracé du treillis de Galois.

Les algorithmes que nous avons implémentés dans ce package sont les algorithmes de Bordat[Bor86], Norris[Nor78], Chein[M.69] et context2lattice [7]. Les versions d'algorithmes de Bordat , Norris et Chein que nous avons implémentés sont ceux proposés par Guénoche [GDP+90b]. Comme nous l'avons souligné, tous ces algorithmes ne construisent pas automatiquement le diagramme de Hasse. C'est la raison pour laquelle nous avons implémenté une méthode de construction du diagramme à partir d'une liste de concepts.

7.3.2 Construction du diagramme de Hasse

L'algorithme de construction du diagramme de Hasse que nous avons implémenté est celui présenté par Valtchev, Missaoui et Lebrun [VML00]. La première étape consiste à trier la liste des concepts par ordre croissant suivant leur intension. Le premier élément à traiter est le supremum du treillis. L'étape suivante est une boucle dans laquelle, pour chaque concept non encore traité, on calcule les concepts dérivant de l'intersection entre l'intension du concept courant et l'intension des concepts obtenus à l'itération

6. Voir fermeture Chap 3
7. context2 lattice est un algorithme conçu et développé par Etienne Cuvelier

précédente. A la fin de chaque itération, le concept ayant l'intension maximal est sélectionné. Cet élément correspond au super-concept courant. Une arête reliant ce super-concept et le concept courant est alors créé. Avant de passer à l'itération suivante, l'ensemble des concepts déjà présents dans le diagramme de Hasse est mis à jour avec l'ajout de ce super-concept.

7.4 Insertion du code C dans un programme R

7.4.1 Motivations

Certaines de ces fonctions s'articulent autour des boucles. Or en R, les boucles sont des instructions qui coûtent cher en terme de temps d'éxécution. La solution face à cela est l'implémentation de ces bouts de codes dans le langage C.
L'incorporation de fonctions C dans un programme R présente plusieurs avantages. L'utilisation des fonctions écrites en langage C améliore la gestion de la mémoire pendant l'exécution des programmes R. De plus les programmes R disposant de fonctions C jouissent d'un temps exécution plus faible. C'est le cas pour les fonctions de génération de treillis de Galois que nous avons implémentées.
Dans la suite de ce paragraphe, nous allons exposer notre expérience sur l'intégration du code C en R. Nous parlerons tout d'abord des outils utilisés, ensuite nous présenterons la procédure adoptée. Nous terminerons en listant quelques problèmes rencontrés ainsi que les solutions adoptées.

7.4.2 Outils

Nous avons programmé sous l'environnement Windows. Pour cela il nous a fallu télécharger et installer respectivement :

1. La console R version 2.11.1
2. l'éditeur de code Tinn-R 2.3.5.2
3. Rtools.exe : fichier qui installe Perl,MinGW. Ce qui permettra la compilation et la génération des DLLs à partir du code C. Après avoir terminer l'exécution de ce fichier, il est important de mettre à jour le PATH en y ajoutant les répertoires bin de ces logiciels. Le répertoire bin de Rtools doit être positionné au début dans les PATH.
4. MikTex car le LaTeX est utilisé pour générer la documentation au format pdf.

5. CodeBlocks comme éditeur de code pour le C.

7.4.3 Procédure

L'ajout d'une fonction C en R se fait en créant une librairie dynamique (DLL) à partir du code C. Cette librairie sera ensuite chargée dans le code R.

7.4.4 Contraintes et particularités des fonctions C

- R ne peut appeler que des fonctions C de type *void*.
- Les arguments des fonctions C appelées en R doivent être des **pointeurs**.
- Les arguments non scalaires (vecteurs, matrices, tableaux) doivent être alloués avant l'appel.
- Les indices commencent à 1 en R, en C ils commencent à 0.
- Il est prudent de passer la taille des vecteurs en argument.
- En R, les matrices sont stockées colonne par colonne ; en C c'est plutôt ligne par ligne. L'élément d'indice *[i,j]* d'une matrice R correspond à l'élément d'indice *[(j-1)*nombre-de-lignes+(i-1)]* dans C ; sachant que, en C les indices commencent à 0.
- Il existe un équivalent C pour chaque type d'objet R. Ainsi, le type *integer* R correspond à *int ** en C.

7.4.5 Marche à suivre

L'incorporation d'un code C en R se déroule comme suit :

1. Compiler le code C pour générer la librairie dynamique.
   - Ouvrir une console *msdos* et se placer dans le répertoire où se trouve le code source C.
   - Saisir la commande **R CMD SHLIB nomfichier.c** ou bien **gcc –shared -o nomfichier.dll nomfichier.c**. La librairie dynamique créée sera nommée *nomfichier.dll*.NB : il n'y a pas d'espace entre les 2 - précédant *shared*.
- Charger le code C compilé en R.
  - A partir de la console R, se placer dans le répertoire contenant la DLL (le fichier .dll).
  - Utiliser la commande *dyn.load("nomfichier.dll")* pour charger le code compilé dans R.

  - Éventuellement, tester à l'aide de la commande *is.loaded* qui prend comme argument le nom entre guillemets d'une fonction C. Si cette section retourne *TRUE* alors tout va bien.
- Appeler dans le programme R les fonctions programmées en C. Cela se fait avec la commande
  ***out*<- .C(*"nomfonction",retour=argument1,argument2,...*)**.
  Dans cet exemple,la variable R *out* contiendra la valeur de retour, qui en C avait été stockée dans la variable. la valeur de retour de la fonction se trouve dans *argument1*. En R, l'accès au contenu de la variable se fera par l'opération ***out$retour***.

7.4.6 Solution à quelques messages d'erreur

Nous présentons ici quelques messages d'erreur obtenus en R et les solutions envisagées :

- **Message :** *C entry point "fonctionC" not in load table*
  Solution : Vérifier que la Dll contient la fonction *fonctionC* et que cette Dll est bien chargée.
- **Message :** (Après exécution de la commande R CMD) *"R n'est pas reconnu comme commande"*
  Solution : Ajouter le répertoire bin de R au PATH
- **Message :**(Apres exécution de la commande R CMD nomfichier.c)
  No rule to make target 'nomfichier.o', needed by 'nomfichier.dll. Stop.
  Solution : Compiler le code C avec la commande :
  gcc –shared -o nomfichier.dll nomfichier.c
- **Message :** l'exécution d'une fonction R contenant un appel de fonction C plante.
  Solution :
  - Vérifier que les types des arguments correspondent en R et en C.
  - Ne passer pas un argument du genre *var$attr* lors d'un appel C (*.C("fct",as.integer(var$attr))*). Il faut mettre cette valeur dans une variable(*v<-var$attr*) et utilisée celle-ci ($.C("fct", as.integer(v))$).
  - Ne pas oublier que les indices en C commencent à 0 et à 1 en R.

7.4.7 Code C dans le package Galois

Les algorithmes complètement implémentés en R s'exécutent dans un temps très raisonnable (1.3 secondes) pour des matrices de petites tailles(10 lignes et 10 colonnes). En appliquant à l'algorithme des matrices de taille un peu plus grande (26 lignes et 42 colonnes), ces algorithmes obtiennent des temps d'exécution très grand, de l'ordre de 30 minutes. Or ces algorithmes

sont prévus pour tourner sur des matrices de très grande taille.
Pour pallier à ce problème, nous avons implémenté quelques boucles en C. Nous l'avons fait pour l'algorithme de Bordat. Cependant, bien que les bouts de code implémentés en C donnaient des temps d'exécution plus petits que ces codes en R, lorsque nous avons intégré ces codes C dans l'algorithme écrit en R, nous avons obtenu un gain de temps de l'ordre de 1%. Car la génération de concepts par l'algorithme de Bordat se fait par une grande boucle. Vu ce résultat pas très satisfaisant ainsi que la structure de l'algorithme, nous avons opté pour l'implémentation entière de l'algorithme de Bordat en C.

Dans ce chapitre, nous avons présenté de manière détaillée le package Galois qui est l'un des principaux éléments dans la réalisation d'Evarist. Ce package sera soumis à la communauté R dans le but d'être intégré dans le CRAN. Dans le chapitre qui suit, nous allons l'appliquer à un exemple afin d'illustrer les résultats que nous pouvons obtenir.

Chapitre 8

Application

Dans ce chapitre, nous allons, à partir d'un exemple, illustrer tous les résultats obtenus. Pour cela, nous allons prendre comme exemple la table de contexte 3.1. Pour le traitement, cette table sera transformée en matrice booléenne. La croix sera remplacée par la variable booléenne *True*, tandis qu'une case sans croix prendra la valeur *False*. La matrice obtenue est :

	aDesPoils	estOvipare	aDesDents	aDesNageoires	aDesPlumes
Antilope	True	False	True	False	False
Sanglier	True	False	True	False	False
Poulet	False	True	False	False	True
Poisson chat	False	True	True	True	False

TABLE 8.1 – Table booléenne de contexte (animals.context)

Nous allons illustrer les différentes fonctions présentées dans le chapitre précédent. Pour cela, nous définissons deux vecteurs de booléens. Le premier que nous nommons *aindex* représentera les attributs sélectionnés. Par exemple si nous considérons les attributs *estOvipare* et *aDesNageoires*, le vecteur *aindex* se présentera comme suit :

aDesPoils	estOvipare	aDesDents	aDesNageoires	aDesPlumes
False	True	False	True	False

TABLE 8.2 – Table booléenne des attributs (*aindex*)

Le second vecteur représentera les objets. Il se nommera *oindex*. Le vecteur représentant les objets *Sanglier* et *Poisson Chat* se de la forme :

Antilope	Sanglier	Poulet	Poisson Chat
False	True	False	True

TABLE 8.3 – Table booléenne des objets (*oindex*)

Nous allons utiliser cet exemple pour illustrer les fonctions présentées au chapitre précédent.

8.1 ext

La fonction *ext* détermine les objets qui partagent certains attributs. Ainsi, pour trouver les animaux qui sont à la fois ovipares et qui ont des nageoires, on utilisera l'appel de fonction *ext(animals.context,aindex)*. Comme le montre la table 8.4 le seul animal ayant ces caractéristiques est le *Poisson Chat*.

Antilope	Sanglier	Poulet	Poisson Chat
False	False	False	True

TABLE 8.4 – Animaux ovipares et qui ont des nageoires

8.2 int

Rapellons que cette fonction a pour but de déterminer les attributs communs à un ensemble d'objets. Si nous voulons connaître quelles sont les caractéristiques que le *Sanglier* et le *Poisson Chat* ont en commun, on exécutera la fonction *int(animals.context,oindex)*. Le résultat nous donne

aDesPoils	estOvipare	aDesDents	aDesNageoires	aDesPlumes
False	False	True	False	False

TABLE 8.5 – Caractéristiques communes du Sanglier et du Poisson Chat

8.3 attrib2concept

Ces deux méthodes ci-dessus nous ont permis de formaliser la construction d'un concept à partir d'un ensemble d'objet. Cette fonction est une implémentation de l'opération X'' définie dans l'Analyse Formelle de Concepts. La construction d'un concept à partir du vecteur *aindex* d'objet se fait par l'appel de fonction *attrib2concept(animals.context,aindex)*. Le résultat obtenu est :
({ Poisson Chat },{ estOvipare, aDesDents, aDesNageoires })

8.4 obj2concept

Comme la fonction *attrib2concept*, la fonction *obj2concept* utilise les fonctions *int* et *ext* pour la construction d'un concept. Cependant, celle-ci construit le concept sur base d'une liste d'objets donnés. Par exemple, l'exécution de la fonction *obj2concept(animals.context,oindex)* permet la construction du concept sur base de l'ensemble d'objets définis dans le vecteur *oindex*. Et nous donne comme résultat :
({ Antilope, Sanglier, Poisson Chat },{ aDesDents })

8.5 Construction du treillis

Dans cette section, le tableau 8.6 nous montre le résultat complet de l'exécution d'un algorithme de générations de concepts. La figure 8.1 nous

Concepts
({ Poulet },{ estOvipare, aDesPlumes })
({ Poisson Chat },{ estOvipare, aDesDents, aDesNageoires })
({ Antilope, Sanglier },{ aDesPoils, aDesDents })
({ Poulet, Poisson Chat },{ estOvipare })
({ Antilope, Sanglier, Poisson Chat },{ aDesDents })
({ Antilope, Sanglier, Poulet, Poisson Chat },{ })
({ },{ aDespoils, estOvipare, aDesDents, aDesNageoires, aDesPlumes })

TABLE 8.6 – Liste des concepts générés

montre le diagramme de Hasse que nous avons déduit de cette liste de concepts

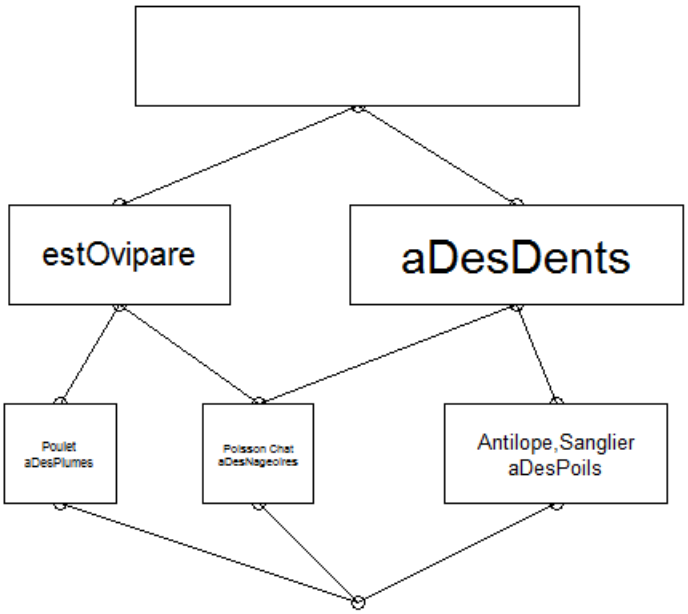


FIGURE 8.1 – Diagramme de Hasse

Conclusion

Dans ce mémoire, nous avons présenté l'outil e-réputation du réseau social Twitter (*Evarist*) développé dans le cadre d'un projet d'analyse de réseaux sociaux en ligne pour l'administration française. L'analyse formelle de concept a été utilisée comme méthode d'analyse, c'est une alternative aux méthodes de clustering. Le développement de cet outil s'est fait dans l'environnement R qui met à la disposition plusieurs méthodes et fonctions utiles pour des calculs matriciels et la représentation graphique. Cet environnement dispose d'une communauté très active proposant divers packages dans des domaines différents. La réalisation de ce projet nous permet d'apporter notre pièrre à l'édifice en proposant un package de construction de Treillis de Galois. Ce package est une implémentation de quelques algorithmes de constructions de Treillis et de représentation graphique du diagrmme de Hasse qui en découle.
Bien qu'il soit encore à l'état de prototype, *Evarist* est un outil simple à utiliser. De manière générale, la saisie du mot qui nous intérèsse ainsi que le nombre de tweets que nous voulons analyser suffisent. Il n'y a pas de syntaxe particulière à conaître comme c'est le cas par exemple pour *Google alert* où il vaut mieux connaître quelques règles de recherche propres à *Google*. De plus, *Evarist* propose aussi un nettoyage de tweets, ce que beaucoup d'outils ne font.
Cependant, nous sommes confrontés à quelques limites qui pourront faire l'objet de futurs travaux. L'environnement de développement R n'est pas interactif. Par conséquent tout se passe sur la console. Les algorithmes tels que nous avons implémenté ont des temps de traitement très élevés lorsqu'ils sont appliqués sur des données de grande taille, surtout lors de la génération du treillis. Or l'outil devra extraire et traiter des données de grande taille. Comme perspective de solution, les auteurs d'Evarist pensent à une génération interactive du treillis. Pour cela, un seuil pourra être défini et dans un premier temps, les niveaux supérieurs du treillis selon présentés. Par la suite, suivant le choix de l'utilisateur, on pourra affiner la présentation d'un niveau.

Table des figures

Liste des tableaux

Bibliographie

[ACM04] M. E. J. Newman Aaron Clauset and Cristopher Moore. Finding community structure in very large networks. *Physical Review E, 70(6) :066111*, 2004.

[AGR] J.B. Angelelli, A. Guénoche, and L. Reboul. Détection de communautés, disjointes ou chevauchantes, dans les réseaux. *Partitionnement de Graphe*.

[Arp10] N. Arpagian. Internet et les réseaux sociaux : outils de contestation et vecteurs d'influence ? *Revue internationale et stratégique*, (2) :97–102, 2010.

[Bai06] S. Baillargeon. Programmation en r : incorporation de code c et création de packages. *Université Laval*, 2006.

[Bak93] H. Bakis. *Les réseaux et leurs enjeux sociaux*. Paris : PUF, coll. "Que sais-je ?", 1993.

[Bar54] John A. Barnes. Class and committees in a norwegian island parish. *Human Relations*, 7 :39,58, 1954.

[Bor86] JP Bordat. Calcul pratique du treillis de galois d'une correspondance. *Math. Sci. Hum*, 96 :31–47, 1986.

[DA05] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 2 :72, 2005.

[DF99] A. Degenne and M. Forsé. *Introducing social networks*. Sage Publications Ltd : London, 1999.

[DM05] L. Donetti and M. A. Muñoz. Improved spectral algorithm for the detection of network communities. *In Modeling cooperative behavior in the social sciences*, 779 :104–107, 2005.

[Dzi08] G. Dziczkowski. Analyse des sentiments : système autonome d'exploration des opinions exprimées dans les critiques cinématographiques. In *Annual Conference (APFA)*, volume 35, page 43, 2008.

[EGBG09] G. Erétéo, F. Gando, M. Buffa, and P. Grohan. Analyse des réseaux sociaux et web sémantique : un état de l'art. *Document ANR nANR-08-cord-011-05*, 2009.

[FF02] B. Fox and C.J. Fox. Efficient stemmer generation. *Information Processing & Management*, 38(4) :547–558, 2002.

[FR91] T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software- Practice and Experience*, 21(11) :1129–1164, 1991.

[Fre79] L.C. Freeman. Centrality in social networks : Conceptual clarification. *Social Networks*, 1 :3215–239, 1979.

[GDP+90a] A. Guenoche, R. DELFORGES, J.L. PETIT, A. LACHENY, E. TEROUANNE, J.P. GINISTI, J.P. DESCLES, B. Henry, V.A.N.M. Iven, A. GUENOCHE, et al. Construction du treillis de galois d'une relation binaire. *Mathématiques et sciences humaines*, 109 :41–53, 1990.

[GDP+90b] A. Guenoche, R. DELFORGES, J.L. PETIT, A. LACHENY, E. TEROUANNE, J.P. GINISTI, J.P. DESCLES, B. Henry, V.A.N.M. Iven, A. GUENOCHE, et al. Construction du treillis de galois d'une relation binaire. *Mathématiques et sciences humaines*, 109 :41–53, 1990.

[GMMM95] R. Godin, G. Mineau, R. Missaoui, and H. Mili. Méthodes de classification conceptuelle basées sur les treillis de galois et applications. *Revue d'intelligence artificielle*, 9(2) :105–137, 1995.

[H+09] A. Harb et al. Détection d'opinion. *Document numérique*, 11(1) :37–61, 2009.

[HDP+08] A. Harb, G. Dray, M. Plantié, P. Poncelet, M. Roche, and F. Trousset. Détection d'opinion : Apprenons les bons adjectifs ! 2008.

[KO01] S. Kuznetsov and S. Obiedkov. Algorithms for the construction of concept lattices and their diagram graphs. *Principles of Data Mining and Knowledge Discovery*, pages 289–300, 2001.

[LEC10] Jean-Paul LECLERCQ. Théorie des graphes. *Cours INFOB321 aux FUNDP*, 2010.

[Lem99] V. Lemieux. *Les réseaux d'acteurs sociaux*. Presses universitaires de France, 1999.

[M.69] Chein M. Algorithme de recherche de sous-matrices premières d'une matrice. *Bull. Math. R. S. Roumanie*, 13, 1969.

[Mal09] Maria Malek. Introduction à l'analyse des réseaux sociaux. 2009.

[Mer03] P. Mercklé. Les réseaux sociaux. les origines de l'analyse des réseaux sociaux. *CNED, ens-lsh*, 2004, 2003.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E, 69(2) :026113*, 2004.

[Nic06] J. Nicot. Thresor de la langue française, tant ancienne que moderne, 1606.

[Nie74] J. Nieminen. On the centrality in a graph. *Scandinavian Journal of Psychology*, 15(1) :332–336, 1974.

[NJI05] E.M.N.P. NJIWOUA. Treillis de concepts et classification supervisée. *Centre de Recherche en Informatique de Lens - CNRS FRE 2499*, 2005.

[NN05] E.M. Nguifo and P. Njiwoua. Treillis de concepts et classification supervisée. *TSI. Technique et science informatiques*, 24(4) :449–488, 2005.

[Nor78] E.M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23(2) :243–250, 1978.

[NR99] L. Nourine and O. Raynaud. A fast algorithm for building lattices. *Information processing letters*, 71(5-6) :199–204, 1999.

[Or05] T. O reilly. What is web 2.0. *Design patterns and business models for the next generation of software*, 30, 2005.

[Or07] T. O reilly. What is web 2.0 : Design patterns and business models for the next generation of software. *Communications and Strategies*, 65 :17, 2007.

[Par10] A.P.P. Paroubek. Le microblogage pour la microanalyse des sentiments et des opinions. *LIMSI - CNRS, Bât. 508 Universite Paris*, 2010.

[PLG11] F. Poulet and B. Le Grand. 4e atelier visualisation et extraction de connaissances. 2011.

[Pon07] Pascal Pons. Détection des communautés dans les grands graphes de terrain. *Thèse de doctorat*, 2007.

[PP11] A. Pak and P. Paroubek. Microblogging for micro sentiment analysis and opinion mining le microblogging pour la micro analyse des sentiments et des opinons. *Traitement Automatique des Langues*, 51, 2011.

[Sco00] J. Scott. Social network analysis, a handbook. *Document ANR nANR-08-cord-011-05*, 2000.

[Sen08] Pierre Senellart. Introduction aux réseaux sociaux sur le web. *Telecom Paris Tech*, page 105, 2008.

[SF09] F. SOMON and A. FRAYSSE. Théorie des graphes. *Mémoire*, 2009.

[SFM04] Vito Latora Santo Fortunato and Massimo Marchiori. Method to find community structures based on information centrality. *Physical Review E, 70(5) :056104*, 2004.

[TBdM03] M. Bouklit T. Bennouas and F. de Montgoller. Un modèle gravitationnel du web. *In 5ème Rencontres Francophones sur les aspects Algorithmiques des Télécommunications (Algotel),Banyuls (France)*, 2003.

[Tor06] Philipe Torloting. Enjeux et perspectives des réseaux sociaux. *Mémoire, Institut Supérieur de Commerce de Paris*, page 44, 2006.

[VML00] P. Valtchev, R. Missaoui, and P. Lebrun. A fast algorithm for building the hasse diagram of a galois lattice. In *Proceedings of the Colloque LaCIM*, pages 293–306. Citeseer, 2000.

[Wes07] M. Wesch. Web 2.0. *The Machine is Using Us. Retrieved January*, 5 :2008, 2007.

[WF94] S. Wasserman and K. Faust. Social network analysis : methods and applications, 1994.

[WT07] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks :[extended abstract]. In *Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276. ACM, 2007.

[Zac77] W. W. Zachary. *An information flow model for conflict and fission in small groups*, volume 33. Journal of Anthropological Research, 1977.

[ZEN04] Emmanuel ZENOU. Localisation topologique, amers visuels et treillis de galois. *Thèse E.N.S Aéronautique de l'Espace*, pages 109–140, 2004.